- Access Control and Identity Management, 3rd Ed. by Mike Chapple. Publisher: Jones and Bartlett Learning. (Sep, 2020).

- Building an Information Security Awareness Program, 1st Ed. by Bill Gardner and Valerie Thomas. Publisher: Syngress. (Aug, 2014).

- Business Continuity and Disaster Recovery Planning for IT Professionals, 2nd Ed. by Susan Snedaker. Publisher: Syngress. (Sep, 2013).

- Cybersecurity Risk Management by Cynthia Brumfield, Brian Haugli. Publisher: Wiley. (Dec, 2021).

- Digital Forensics and Incident Response, 2nd Ed. by Gerard Johansen. Publisher: Packt Publishing. (Jan, 2020).

- Disaster Recovery, Crisis Response, and Business Continuity: A Management Desk Reference by Jamie Watters, Janet Watters. Publisher: Apress. (Dec, 2013).

- Distributed Denial of Service (DDoS) by Eric Chou, Rich Groves. Publisher: O’Reilly Media, Inc. (Apr, 2018).

- Foundations of Information Security: A Straightforward Introduction by Jason Andress. Publisher: William Pollock. (Oct, 2019).

- Fundamentals of Information Systems Security, 4th Ed. by David Kim, Michael G. Solomon. Publisher: Jones & Bartlett Publishers. (Nov, 2021).

- Information Assurance Handbook: Effective Computer Security and Risk Management Strategies, 1st Ed. by Corey Schou and Steven Hernandez. Publisher: McGraw-Hill Education. (Sep, 2014).

- Information Security Policies, Procedures, and Standards: A Practitioner's Reference by Dogulas J. Landoll. Publisher: Auerbach Publications. (Mar, 2017).

- ISC2 Code of Ethics by ISC2. (Dec, 2023).

- Mastering Windows Security and Hardening by Mark Dunkerley, Matt Tumbarello. Publisher: Packt Publishing. (Aug, 2022).

- Modern Cryptography for Security Professionals, 1st Ed. by Lisa Bock. Publisher: Packt Publishing. (Jun, 2021).

- Network Security, Firewalls, and VPNs, 3rd Edition by J. Michael Stewart, Denise Kinsey. Publisher: Jones & Bartlett Learning. (Oct, 2020).

- Networking Fundamentals, 1st Ed. by Gordon Davies. Publisher: Packt Publishing. (Dec, 2019).

- Network Security Strategies by Aditya Mukherjee. Publisher: Packt Publishing. (Nov, 2020).

- NIST SP 800-53, Rev. 5, Security and Privacy Controls for Information Systems and Organizations by Joint Task Force Transformation Initiative. (Sep, 2020).

- NIST SP 800-61, Rev. 2, Computer Security Incident Handling Guide by Paul Cichonski, Tom Millar, Tim Grance, Karen Scarfone. (Aug, 2012).

- Security Policies and Implementation Issues, 3rd Ed. by Robert Johnson and Chuck Easttom. Publisher: Bartlett Learning. (Oct, 2020).

- The Complete Guide to Physical Security by Paul R. Baker and Daniel J. Benny. Publisher: Auerbach Publications. (Apr, 2016).
- A Practical Guide to TPM 2.0: Using the New Trusted Platform Module in the New Age of Security by Will Arthur, David Challener. Publisher: Apress. (Jan, 2015).

- Access Control and Identity Management, 3rd Ed. by Mike Chapple. Publisher: Jones and Bartlett Learning. (Sep, 2020).

- A Technical Guide to IPSec Virtual Private Networks by James S. Tiller. Publisher: Auerbach Publications. (Jul, 2017).

- Applied Cryptography: Protocols, Algorithms and Source Code in C, 20th Anniversary Ed. by Bruce Schneier. Publisher: Wiley. (Mar, 2015).

- Building an Information Security Awareness Program, 1st Ed. by Bill Gardner and Valerie Thomas. Publisher: Syngress. (Aug, 2014).

- Business Continuity and Disaster Recovery Planning for IT Professionals, 2nd Ed. by Susan Snedaker. Publisher: Syngress. (Sep, 2013).

- Cryptography and Network Security Principles and Practice, 6th Ed. by William Stallings. Publisher: Pearson. (Mar, 2013).

- Cybersecurity Incident Response: How to Contain, Eradicate, and Recover from Incidents by Eric C. Thompson. Publisher: Apress. (Sep, 2018).

- Digital Forensics and Incident Response, 2nd Ed. by Gerard Johansen. Publisher: Packt Publishing. (Jan, 2020).

- Foundations of Information Security: A Straightforward Introduction by Jason Andress. Publisher: William Pollock. (Oct, 2019).

- Fundamentals of Information Systems Security, 4th Ed. by David Kim, Michael G. Solomon. Publisher: Jones & Bartlett Publishers. (Nov, 2021).

- Identity Attack Vectors: Implementing an Effective Identity and Access Management Solution by Darran Rolls, Morey J. Haber. Publisher: Apress. (Dec, 2019).

- Identity and Access Management: Business Performance Through Connected Intelligence, 1st Ed. by Ertem Osmanoglu. Publisher: Syngress. (Nov, 2013).

- Information Risk Management: A Practitioner's Guide by David Sutton. Publisher: BCS, The Chartered Institute for IT. (Nov, 2014).

- Introduction to Computer Networks and Cybersecurity, 1st Ed. by J. Chwan-Hwa Wu, David Irwin. Publisher: CRC Press. (Apr, 2016).

- ISC2 Code of Ethics by ISC2. (Dec, 2023).

- Logging and Log Management by A. Chuvakin, K. Schmidt. Publisher: Syngress. (Dec, 2012).

- Network Security, Firewalls, and VPNs, 3rd Edition by J. Michael Stewart, Denise Kinsey. Publisher: Jones & Bartlett Learning. (Oct, 2020).

- Networking Fundamentals, 1st Ed. by Gordon Davies. Publisher: Packt Publishing. (Dec, 2019).

- NIST SP 800-61, Rev. 2, Computer Security Incident Handling Guide by Paul Cichonski, Tom Millar, Tim Grance, Karen Scarfone. (Aug, 2012).

- Official (ISC)² SSCP CBK Reference, 5th Ed. by Mike Wills. Publisher: Sybex. (Dec, 2019).

- Practical Cloud Security: A Guide for Secure Design and Deployment by Chris Dotson. Publisher: O'Reilly Media. (Mar, 2019).

- A Practical Guide to TPM 2.0: Using the New Trusted Platform Module in the New Age of Security by Will Arthur, David Challener. Publisher: Apress. (Jan, 2015).

- Access Control, Authentication, and Public Key Infrastructure (Information Systems Security & Assurance), 1st Ed. by Bill Ballad, Tricia Ballad, Erin Banks. Publisher: Jones & Bartlett Learning. (Oct, 2010).

- Agile Application Security by Laura Bell, Rich Smith, Michael Brunton-Spall, Jim Bird. Publisher: O’Reilly Media, Inc. (Jun, 2017).

- Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS) by Michael Kavis. Publisher: Wiley. (Jan, 2014).

- Asset Attack Vectors: Building Effective Vulnerability Management Strategies to Protect Organizations by Morey J. Haber, Brad Hibbert. Publisher: Apress. (Jun, 2018).

- Business Continuity and Disaster Recovery Planning for IT Professionals, 2nd Ed. by Susan Snedaker. Publisher: Syngress. (Sep, 2013).

- Computer and Information Security Handbook, 3rd Ed. by John Vacca. Publisher: Morgan Kaufmann. (May, 2017).

- Computer Security: Art and Science, 2nd Ed. by Matt Bishop. Publisher: Addison-Wesley Professional. (Nov, 2018).

- Core Software Security: Security at the Source by Anmol Misra, James F. Ransome. Publisher: Auerbach Publications. (Oct, 2018).

- Data Center Handbook, 2nd Ed. by Hwaiyu Geng. Publisher: Wiley. (May, 2021).

- Developing Cybersecurity Programs and Policies, 3rd Ed. by Omar Santos, Sari Greene. Publisher: Pearson IT Certification. (Aug, 2018).

- Disaster Recovery, Crisis Response, and Business Continuity: A Management Desk Reference by Jamie Watters, Janet Watters. Publisher: Apress. (Dec, 2013).

- Distributed Denial of Service (DDoS) by Eric Chou, Rich Groves. Publisher: O’Reilly Media, Inc. (Apr, 2018).

- DomainKeys Identified Mail (DKIM) Signature by Murray Kucherawy, Dave Crocker, Tony Hansen. Publisher: IETF. (Sep, 2011).

- Ethical Hacking and Penetration Testing Guide by Rafay Baloch. Publisher: Auerbach Publications. (Sep, 2017).

- Federated Identity Primer, 1st Ed. by Derrick Rountree. Publisher: Syngress. (Dec, 2012).

- Foundations of Information Security: A Straightforward Introduction by Jason Andress. Publisher: William Pollock. (Oct, 2019).

- Fundamental Practices for Secure Software Development: Essential Elements of a Secure Development Lifecycle Program, 3rd Ed. by Toni Rice. Publisher: SAFECode. (Mar, 2018).

- Fundamentals of Information Systems Security, 4th Ed. by David Kim, Michael G. Solomon. Publisher: Jones & Bartlett Publishers. (Nov, 2021).

- EU General Data Protection Regulation (GDPR) by European Parliament. Publisher: European Parliament and Council of the European Union. (May, 2016).

- Identity and Access Management: Business Performance Through Connected Intelligence, 1st Ed. by Ertem Osmanoglu. Publisher: Syngress. (Nov, 2013).

- Identity Attack Vectors: Implementing an Effective Identity and Access Management Solution by Darran Rolls, Morey J. Haber. Publisher: Apress. (Dec, 2019).

- Information Assurance Handbook: Effective Computer Security and Risk Management Strategies, 1st Ed. by Corey Schou and Steven Hernandez. Publisher: McGraw-Hill Education. (Sep, 2014).

- Information Security Handbook by Darren Death. Publisher: Packt Publishing. (Dec, 2017).

- Information Security Policies, Procedures, and Standards: A Practitioner’s Reference by Dogulas J. Landoll. Publisher: Auerbach Publications. (Mar, 2017).

- Introduction to Computer Networks and Cybersecurity, 1st Ed. by J. Chwan-Hwa Wu, David Irwin. Publisher: CRC Press. (Apr, 2016).

- ISC2 Code of Ethics by ISC2. (Dec, 2023).

- IT Auditing Using Controls to Protect Information Assets, 3rd Edition by Mike Kegerreis, Mike Schiller, Chris Davis. Publisher: McGraw-Hill Education. (Oct, 2019).

- Network Security, Firewalls, and VPNs, 3rd Edition by J. Michael Stewart, Denise Kinsey. Publisher: Jones & Bartlett Learning. (Oct, 2020).

- Network Vulnerability Assessment: Identify Security Loopholes in Your Network's Infrastructure by Sagar Rahalkar. Publisher: Packt Publishing. (Aug, 2018).

- Networking Fundamentals, 3rd Ed. by Chuck Easttom, Richard M. Roberts. Publisher: Goodheart-Willcox. (Sep, 2018).

- NIST SP 800-30, Rev. 1, Guide for Conducting Risk Assessments by Joint Task Force Transformation Initiative. (Sep, 2012).

- NIST SP 800-34 Rev. 1, Contingency Planning Guide for Federal Information Systems by Marianne Swanson, Pauline Bowen, Amy Wohl Phillips, Dean Gallup, David Lynes. (May, 2010).

- NIST SP 800-41, Revision 1, Guidelines on Firewalls and Firewall Policy by Karen Scarfone, Paul Hoffman. (Sep, 2009).

- NIST SP 800-53, Rev. 5, Security and Privacy Controls for Information Systems and Organizations by Joint Task Force Transformation Initiative. (Sep, 2020).

- NIST SP 800-63-3, Digital Identity Guidelines: Enrollment and Identity Proofing by Paul A. Grassi, James L. Fenton, Naomi B. Lefkovitz, Jamie M. Danker, Yee-Yin Choong, Kristen K. Greene, Mary F. Theofanos. (Jun, 2017).

- NIST SP 800-77, Revision 1, Guide to IPsec VPNs by Elaine Barker, Quynh Dang, Sheila Frankel, Karen Scarfone, Paul Wouters. Publisher: NIST. (Jun, 2020).

- NIST SP 800-88, Guidelines for Media Sanitization by Richard Kissel, Andrew Regenscheid, Matthew Scholl, Kevin Stine. (Dec, 2014).

- NIST SP 800-95, Guide to Secure Web Services by Anoop Singhal, Theodore Winograd, Karen Scarfone. (Aug, 2007).

- NIST SP 800-115, Technical Guide to Information Security Testing and Assessment by Karen Scarfone, Murugiah Souppaya, Amanda Cody, Angela Orebaugh. (Sep, 2008).

- NIST SP 800-122, Guide to Protecting the Confidentiality of Personally Identifiable Information (PII) by Erika McCallister, Tim Grance, Karen Scarfone. (Apr, 2010).

- NIST SP 800-128, Guide for Security-Focused Configuration Management of Information Systems by Arnold Johnson, Kelley Dempsey, Ron Ross, Sarbari Gupta, Dennis Bailey. (Aug, 2011).

- NIST SP 800-137, Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations by Kelley Dempsey, Nirali Shah Chawla, Arnold Johnson, Ronald Johnston, Alicia Clay Jones, Angela Orebaugh, Matthew Scholl, Kevin Stine. (Sep, 2011).

- NIST SP 800-160, Vol. 1, Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems by Ron Ross, Michael McEvilley, Janet Carrier Oren. (Mar, 2018).

- Official (ISC)² Guide to the CISSP CBK, 5th Ed. by John Warsinske, Mark Graff, Kevin Henry, Christopher Hoover, Ben Malisow, Sean Murphy, C. Paul Oakes, George Pajari, Jeff T. Parker, David Seidl and Mike Vasquez. Publisher: Wiley. (May, 2019).

- OWASP Testing Guide, Release 4.0 by Matteo Meucci, Andrew Muller. Publisher: OWASP. (Dec, 2014).

- Physical Security and Safety by Jeffrey Dingle, Bobby E. Ricks, Truett A, Ricks. Publisher: CRC Press. (Oct, 2014).

- Practical Cloud Security: A Guide for Secure Design and Deployment by Chris Dotson. Publisher: O'Reilly Media. (Mar, 2019).

- Practical Internet of Things Security, 2nd Ed. by Brian Russel, Drew Van Duren. Publisher: Packt Publisher. (Nov, 2018).

- Securing Open Source Libraries by Guy Podjarny. Publisher: O’Reilly Media, Inc. (Nov, 2017).

- Securing VoIP, 1st Ed. by Regis Bates. Publisher: Syngress. (Nov, 2014).

- Security Policies and Implementation Issues, 3rd Ed. by Robert Johnson and Chuck Easttom. Publisher: Bartlett Learning. (Oct, 2020).

- Security Risk Assessment: Managing Physical and Operational Security by John M. White. Publisher: Butterworth-Heinemann. (Jul, 2014).

- Solving Identity Management in Modern Applications: Demystifying OAuth 2.0, OpenID Connect, and SAML 2.0 by Abhishek Hingnikar, Yvonne Wilson. Publisher: Apress. (Nov, 2022).

- The Complete Guide to Physical Security by Paul R. Baker and Daniel J. Benny. Publisher: Auerbach Publications. (Apr, 2016).

- The Disaster Recovery Handbook, 3rd Ed. by Michael Wallace, Lawrence Webber. Publisher: AMACOM. (Dec, 2017).

- Threat Modeling: Designing for Security, 1st Ed. by Adam Shostack. Publisher: Wiley. (Feb, 2014).

- Web Application Security: Exploitation and Countermeasures for Modern Web Applications by Andrew Hoffman. Publisher: O’Reilly Media, Inc. (Mar, 2020).

- Zero Trust Networks: Building Secure Systems in Untrusted Networks by Evan Gilman, Doug Barth. Publisher: O'Reilly. (Jul, 2017).

- Architecting Cloud Computing Solutions by Kevin L. Jackson and Scott Goessling. Publisher: Packt Publishing. (May, 2018).

- Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS) by Michael Kavis. Publisher: Wiley. (Jan, 2014).

- Business Continuity and Disaster Recovery Planning for IT Professionals, 2nd Ed. by Susan Snedaker. Publisher: Syngress. (Sep, 2013).

- Cloud Computing Design Patterns by Thomas Erl, Robert Cope, Amin Naserpour. Publisher: Prentice Hall. (Mar, 2017).

- Cloud Security Handbook by Eyal Estrin. Publisher: Packt Publishing. (Apr, 2022).

- Data Governance: The Definitive Guide by Evren Eryurek, Uri Gilad, Valliappa Lakshmanan, Anita Kibunguchy-Grant, Jessi Ashdown. Publisher: O'Reilly Media, Inc. (Mar, 2021).

- EU General Data Protection Regulation (GDPR) by European Parliament. Publisher: European Parliament and Council of the European Union. (May, 2016).

- Fundamental Practices for Secure Software Development. Publisher: SAFECode. (Mar, 2018).

- Guide to Privacy and Security of Electronic Health Information. Publisher: HealthIT.gov. (Apr, 2015).

- Incident Response in the Age of Cloud: Techniques and best practices to effectively respond to cybersecurity incidents by Erdal Ozkaya. Publisher: Packt Publishing. (Feb, 2021).

- Information Security Handbook by Darren Death. Publisher: Packt Publishing. (Dec, 2017).

- NIST SP 800-125, Guide to Security for Full Virtualization Technologies by Karen Scarfone, Murugiah Souppaya, Paul Hoffman. (Jan, 2011).

- Official (ISC)² Guide to the CCSP CBK, 3rd Ed. by Leslie Fife, Aaron Kraus, Bryan Lewis. Publisher: Sybex. (July, 2021).

- Practical Cloud Security: A Guide for Secure Design and Deployment by Chris Dotson. Publisher: O'Reilly Media. (Mar, 2019).

- Security Guidance for Critical Areas of Focus in Cloud Computing v4.0 by Rich Mogull, James Arlen, Adrian Lane, Gunnar Peterson, Mike Rothman, David Mortman. Publisher: Cloud Security Alliance. (Jul, 2017).

- Security, Privacy, and Digital Forensics in the Cloud by Lei Chen, Hassan Takabi, Nhien-An Le-Khac. Publisher: Wiley. (Apr, 2019).
- Information Security Risk Management for ISO 27001/ISO 27002, 3rd Ed. by Alan Calder, Steve Watkins. Publisher: IT Governance Publishing. (Aug, 2019).

- ISO 27001/ISO 27002, A Pocket Guide, 2nd Ed. By Alan Calder. Publisher: IT Governance Publishing. (Oct, 2013).

- NIST FIPS-199, Standards for Security Categorization of Federal Information and Information Systems by U.S. Dept. of Commerce. (Feb, 2004).

- NIST SP 800-18, Rev. 1, Guide for Developing Security Plans for Federal Information Systems by Marianne Swanson, Joan Hash, Pauline Bowen. (Feb, 2006).

- NIST SP 800-30, Rev. 1, Guide for Conducting Risk Assessments by Joint Task Force Transformation Initiative. (Sep, 2012).

- NIST SP 800-37, Rev. 2, Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy by Joint Task Force Transformation Initiative. (Dec, 2018).

- NIST SP 800-39, Managing Information Security Risk: Organization, Mission, and Information System View by Joint Task Force Transformation Initiative. (Mar, 2011).

- NIST SP 800-53, Rev. 5, Security and Privacy Controls for Information Systems and Organizations by Joint Task Force Transformation Initiative. (Sep, 2020).

- NIST SP 800-53A, Rev. 5, Assessing Security and Privacy Controls in Information Systems and Organizations by Joint Task Force Transformation Initiative. (Jan, 2022).

- NIST SP 800-53B, Control Baselines for Information Systems and Organizations by Joint Task Force Transformation Initiative. (Sep, 2020).

- NIST SP 800-60, Vol. 1, Rev. 1, Guide for Mapping Types of Information and Information Systems to Security Categories by Kevin Stine, Rich Kissel, William C. Barker, Jim Fahlsing, Jessica Gulick. (Aug, 2008).

- NIST SP 800-70, Rev. 4, National Checklist Program for IT Products: Guidelines for Checklist Users and Developers by Stephen D. Quinn, Murugiah Souppaya, Melanie Cook, Karen Scarfone. (Feb, 2018).

- NIST SP 800-88, Guidelines for Media Sanitization by Richard Kissel, Andrew Regenscheid, Matthew Scholl, Kevin Stine. (Dec, 2014).

- NIST SP 800-137, Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations by Kelley Dempsey, Nirali Shah Chawla, Arnold Johnson, Ronald Johnston, Alicia Clay Jones, Angela Orebaugh, Matthew Scholl, Kevin Stine. (Sep, 2011).
- A Guide to Building Secure Web Applications and Web Services 2.0 Black Hat Ed. by Abraham Kang, Adrian Wiesmann, et al. Publisher: OWASP. (Jul, 2005).

- A Practical Guide to TPM 2.0: Using the New Trusted Platform Module in the New Age of Security by Will Arthur, David Challener. Publisher: Apress. (Jan, 2015).

- Access Control, Authentication, and Public Key Infrastructure, 2nd Ed. by Mike Chapple, Bill Ballad, Tricia Ballad, Erin Banks. Publisher: Jones & Bartlett Learning. (Jul, 2013).

- Agile Application Security by Laura Bell, Rich Smith, Michael Brunton-Spall, Jim Bird. Publisher: O’Reilly Media, Inc. (Jun, 2017).

- Applied Cryptography: Protocols, Algorithms and Source Code in C, 20th Anniversary Ed. by Bruce Schneier. Publisher: Wiley. (Mar, 2015).

- CMMI for Development: Implementation Guide by Mukund Chaudhary, Abhishek Chopra. Publisher: Apress. (Dec, 2016).

- Computer Security: Art and Science, 2nd Ed. by Matt Bishop. Publisher: Addison-Wesley Professional. (Nov, 2018).

- Core Software Security: Security at the Source by Anmol Misra, James F. Ransome. Publisher: Auerbach Publications. (Oct, 2018).

- Cybersecurity - Attack and Defense Strategies, 2nd Ed. by Erdal Ozkaya and Yuri Diogenes. Publisher: Packt Publishing. (Dec, 2019).

- Enterprise Software Security: A Confluence of Disciplines by Kenneth R. van Wyk, Mark G. Graff, Dan S. Peters, Diana L. Burley. Publisher: Addison-Wesley Professional. (Dec, 2014).

- Hacker Techniques, Tools, and Incident Handling, 2nd Ed. by Sean-Philip Oriyano. Publisher: Jones & Bartlett Learning. (Aug, 2013).

- Hands-On Security in DevOps by Tony Hsu. Publisher: Packt Publishing. (Jul, 2018).

- Improper Error Handling by Jeremy Ferragamo, Wichers, Jim Bird. Publisher: OWASP. (Dec, 2021).

- Information Security: Principles and Practices, 2nd Ed. by Mark S. Merkow, Jim Breithaupt. Publisher: Pearson IT Certification. (Jun, 2014).

- IT Release Management: A Hands-on Guide by Dave Howard. Publisher: CRC Press. (Apr, 2016).

- IT Security Risk Control Management: An Audit Preparation Plan by Raymond Pompon. Publisher: Apress. (Sep, 2016).

- Lessons Learned in Software Testing: A Context-Driven Approach by Bret Pettichord, Cem Kaner, James Marcus Bach. Publisher: Wiley. (Dec, 2001).

- Logging and Log Management by A. Chuvakin, K. Schmidt. Publisher: Syngress. (Dec, 2012).

- Mastering the Requirements Process: Getting Requirements Right v3.0 by S. Robertson, J. Robertson. Publisher: Addison-Wesley Professional. (Aug, 2012).

- NIST SP 800-37, Rev. 2, Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy by Joint Task Force Transformation Initiative. (Dec, 2018).

- NIST SP 800-53, Rev. 5, Security and Privacy Controls for Information Systems and Organizations by Joint Task Force Transformation Initiative. (Sep, 2020).

- NIST SP 800-60, Vol. 1, Rev. 1, Guide for Mapping Types of Information and Information Systems to Security Categories by Kevin Stine, Rich Kissel, William C. Barker, Jim Fahlsing, Jessica Gulick. (Aug, 2008).

- NIST SP 800-88, Guidelines for Media Sanitization by Richard Kissel, Andrew Regenscheid, Matthew Scholl, Kevin Stine. (Dec, 2014).

- NIST IR 7622, Notional Supply Chain Risk Management Practices for Federal Information Systems by Jon Boyens, Celia Paulsen, Nadya Bartol, Stephany A. Shankles, Rama Moorthy. (Oct, 2012).

- Official (ISC)² Guide to the CSSLP, 2nd Ed. by Mano Paul. Publisher: Auerbach Publications. (Aug, 2013).

- OWASP Testing Guide, Release 4.0 by Matteo Meucci, Andrew Muller. Publisher: OWASP. (Dec, 2014).

- Penetration Testing: A Survival Guide by W. Halton, B. Weaver, J. Ansari, S. Kotipalli, M. Imran. Publisher: Packt Publishing. (Jan, 2017).

- Security Risk Management by Evan Wheeler. Publisher: Syngress. (Apr, 2011).

- Software Testing Foundations: A Study Guide for the Certified Tester Exam, 4th Ed. by Andreas Spillner. Publisher: Rocky Nook. (Feb, 2014).

- Web Application Firewalls by Chad Russell. Publisher: O’Reilly Media, Inc. (Apr, 2018).
- Agile Application Lifecycle Management: Using DevOps to Drive Process Improvement, 1st Ed. by Bob Aiello, Leslie Sachs. Publisher: Addison-Wesley Professional. (Jun, 2016).

- Agile Application Security by Laura Bell, Rich Smith, Michael Brunton-Spall, Jim Bird. Publisher: O'Reilly Media, Inc. (Jun, 2017).

- Architecting Secure Software Systems, 1st Ed. by Asoke Talukder, Manish Chaitanya. Publisher: Auerbach Publications. (Sep, 2019).

- Applied Cryptography: Protocols, Algorithms and Source Code in C, 20th Anniversary Ed. by Bruce Schneier. Publisher: Wiley. (Mar, 2015).

- Beginning Database Design Solutions by Rod Stephens. Publisher: Jossey-Bass. (Nov, 2008).

- Business Continuity and Disaster Recovery Planning for IT Professionals, 2nd Ed. by Susan Snedaker. Publisher: Syngress. (Sep, 2013).

- Cloud Storage Security: A Practical Guide by Aaron Wheeler, Michael Winburn. Publisher: Elsevier. (Jul, 2015).

- Common Criteria for Information Technology Security Evaluation, Version 3.1 Rev. 5 by Mead, N. Publisher: Carnegie. (Apr, 2017).

- Data Center Handbook, 2nd Ed. by Hwaiyu Geng. Publisher: Wiley. (May, 2021).

- Disaster Recovery and Business Continuity, 3rd Ed. by B.S. Thejandra. Publisher: IT Governance Publishing. (Jan, 2014).

- Enterprise Security Architecture: A Business-Driven Approach, 1st Ed. by John Sherwood. Publisher: CRC Press. (Nov, 2015).

- Identity and Access Management: Business Performance Through Connected Intelligence, 1st Ed. by Ertem Osmanoglu. Publisher: Syngress. (Nov, 2013).

- Information Security Management Handbook, Vol. 6, 6th Ed. by Harold F. Tipton and Micki Krause Nozaki. Publisher: Auerbach Publications. (Apr, 2016).

- Information Security Management Handbook, Vol. 7, 6th Ed. by Richard O'Hanley, James Tiller. Publisher: Auerbach Publications. (Aug, 2013).

- NIST SP 800-34 Rev. 1, Contingency Planning Guide for Federal Information Systems by Marianne Swanson, Pauline Bowen, Amy Wohl Phillips, Dean Gallup, David Lynes. (May, 2010).

- NIST SP 800-57, Rev. 5, Recommendation for Key Management: Part 1 – General by Elaine Barker. Publisher: NIST. (May, 2020) .

- NIST SP 800-61, Rev. 2, Computer Security Incident Handling Guide by Paul Cichonski, Tom Millar, Tim Grance, Karen Scarfone. (Aug, 2012).

- NIST SP 800-63-3, Digital Identity Guidelines: Enrollment and Identity Proofing by Paul A. Grassi, James L. Fenton, Naomi B. Lefkovitz, Jamie M. Danker, Yee-Yin Choong, Kristen K. Greene, Mary F. Theofanos. (Jun, 2017).

- NIST SP 800-115, Technical Guide to Information Security Testing and Assessment by Karen Scarfone, Murugiah Souppaya, Amanda Cody, Angela Orebaugh. (Sep, 2008).

-]NIST SP 800-125, Guide to Security for Full Virtualization Technologies by Karen Scarfone, Murugiah Souppaya, Paul Hoffman. (Jan, 2011).

- NIST SP 800-162, Guide to Attribute Based Access Control (ABAC) Definition and Considerations by Vincent Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone. (Jan, 2014).

- Official (ISC)² Guide to the ISSAP CBK, 2nd Ed. by Adam Gordon. Publisher: Auerbach Publications. (Jan, 2017).

- Payment Card Industry Data Security Standards, Requirements and Security Assessment Procedures, Version 3.2.1 by PCI Security Standards Council. Publisher: PCI Security Standards Council, LLC. (May, 2018).

- Practical Internet of Things Security, 2nd Ed. by Brian Russel, Drew Van Duren. Publisher: Packt Publisher. (Nov, 2018).

- SABSA Executive Summary by SABSA. Publisher: The SABSA Institute. (Dec, 2021).

- Secure Coding in C and C++, 2nd Ed. by Robert Seacord. Publisher: Addison-Wesley Professional. (Apr, 2013).

- Security Patterns in Practice: Designing Secure Architectures Using Software Patterns by Eduardo Fernandez-Buglioni. Publisher: Wiley. (May, 2013).

- Security Guidance for Critical Areas of Focus in Cloud Computing v4.0 by Rich Mogull, James Arlen, Adrian Lane, Gunnar Peterson, Mike Rothman, David Mortman. Publisher: Cloud Security Alliance. (Jul, 2017).
- A Guide to the Project Management Body of Knowledge (PMBOK Guide), 7th Ed. by Project Management Institute. Publisher: Project Management Institute. (Aug, 2021).

- INCOSE Systems Engineering Handbook by Walden. Publisher: Wiley. (Jul, 2015).

- Information Assurance Technical Framework 3.1 by National Security Agency Information Assurance Solutions Technical Directors. (Sep, 2002).

- ISO/IEC 15408 Common Criteria for Information Technology Security Evaluation by ISO/IEC. Publisher: National Information Assurance Partnership. (Dec, 2017).

- NIST SP 800-30, Rev. 1, Guide for Conducting Risk Assessments by Joint Task Force Transformation Initiative. (Sep, 2012).

- NIST SP 800-37, Rev. 2, Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy by Joint Task Force Transformation Initiative. (Dec, 2018).

- NIST SP 800-39, Managing Information Security Risk: Organization, Mission, and Information System View by Joint Task Force Transformation Initiative. (Mar, 2011).

- NIST SP 800-40, Rev. 3, Guide to Enterprise Patch Management Technologies Murugiah Souppaya, Karen Scarfone. (Jul, 2013).

- NIST SP 800-53, Rev. 5, Security and Privacy Controls for Information Systems and Organizations by Joint Task Force Transformation Initiative. (Sep, 2020).

- NIST SP 800-88, Guidelines for Media Sanitization by Richard Kissel, Andrew Regenscheid, Matthew Scholl, Kevin Stine. (Dec, 2014).

- NIST SP 800-115, Technical Guide to Information Security Testing and Assessment by Karen Scarfone, Murugiah Souppaya, Amanda Cody, Angela Orebaugh. (Sep, 2008).

- NIST SP 800-160, Vol. 1, Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems by Ron Ross, Michael McEvilley, Janet Carrier Oren. (Mar, 2018).

- NIST SP 800-161, Supply Chain Risk Management Practices for Federal Information System and Organizations by Jon Boyens, Celia Paulsen, Rama Moorthy, Nadya Bartol. (Apr, 2015).
- A Guide to the Project Management Body of Knowledge (PMBOK Guide), 7th Ed. by Project Management Institute. Publisher: Project Management Institute. (Aug, 2021).

- Auditing IT Infrastructures for Compliance, 2nd Ed. by Martin Weiss. Publisher: Jones & Bartlett Publishers. (Jul, 2015).

- Business Continuity and Disaster Recovery Planning for IT Professionals, 2nd Ed. by Susan Snedaker. Publisher: Syngress. (Sep, 2013).

- Digital Forensics and Incident Response, 2nd Ed. by Gerard Johansen. Publisher: Packt Publishing. (Jan, 2020).

- Disaster Recovery, Crisis Response, and Business Continuity: A Management Desk Reference by Jamie Watters, Janet Watters. Publisher: Apress. (Dec, 2013).

- Disaster Recovery Planning: For Computers and Communication Resources by Jon Toigo. Publisher: Wiley. (Jan, 1996).

- Information Security Management Handbook, Vol. 6, 6th Ed. by Harold F. Tipton and Micki Krause Nozaki. Publisher: Auerbach Publications. (Apr, 2016).

- ISC2 Code of Ethics by ISC2. (Dec, 2023).

- Incident Response & Computer Forensics, 3rd Ed. by Jason Luttgens, Matthew Pepe, Kevin Mandia. Publisher: McGraw-Hill Osborne Media. (Aug, 2014).

- IT Auditing Using Controls to Protect Information Assets, 3rd Edition by Mike Kegerreis, Mike Schiller, Chris Davis. Publisher: McGraw-Hill Education. (Oct, 2019).

- NIST SP 800-30, Rev. 1, Guide for Conducting Risk Assessments by Joint Task Force Transformation Initiative. (Sep, 2012).

- NIST SP 800-34 Rev. 1, Contingency Planning Guide for Federal Information Systems by Marianne Swanson, Pauline Bowen, Amy Wohl Phillips, Dean Gallup, David Lynes. (May, 2010).

- NIST SP 800-37, Rev. 2, Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy by Joint Task Force Transformation Initiative. (Dec, 2018).

- NIST SP 800-39, Managing Information Security Risk: Organization, Mission, and Information System View by Joint Task Force Transformation Initiative. (Mar, 2011).

- NIST SP 800-53, Rev. 5, Security and Privacy Controls for Information Systems and Organizations by Joint Task Force Transformation Initiative. (Sep, 2020).

- NIST SP 800-55, Rev. 1, Performance Measurement Guide for Information Security by Elizabeth Chew, Marianne Swanson, Kevin Stine, Nadya Bartol, Anthony Brown, Will Robinson. (Jul, 2008).

- NIST SP 800-61, Rev. 2, Computer Security Incident Handling Guide by Paul Cichonski, Tom Millar, Tim Grance, Karen Scarfone. (Aug, 2012).

- NIST SP 800-128, Guide for Security-Focused Configuration Management of Information Systems by Arnold Johnson, Kelley Dempsey, Ron Ross, Sarbari Gupta, Dennis Bailey. (Aug, 2011).

- NIST SP 800-160, Vol. 1, Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems by Ron Ross, Michael McEvilley, Janet Carrier Oren. (Mar, 2018).

- Official (ISC)² Guide to the ISSMP CBK by Joseph Steinberg and Harold F. Tipton. Publisher: Auerbach Publications. (Apr, 2016).

- Security Operations Center: Building, Operating, and Maintaining your SOC by Gary McIntyre, Joseph Muniz, Nadhem AlFardan. Publisher: Cisco Press. (Nov, 2015).

- The Disaster Recovery Handbook, 3rd Ed. by Michael Wallace, Lawrence Webber. Publisher: AMACOM. (Dec, 2017).

- Threat Modeling: Designing for Security, 1st Ed. by Adam Shostack. Publisher: Wiley. (Feb, 2014).

nline Training

Official ISC2 Certified in Cybersecurity (CC) eTextbook

eTextbook Description

< Return to Listing

The Official ISC2 Certified in Cybersecurity (CC) eTextbook is your go-to learning resource as you prepare for the CC exam. It provides a comprehensive review of the topics covered in the Official ISC2 Training Course and will help you navigate key cybersecurity concepts as you build confidence for exam day.

Price: $24.95

Language:

English

Japanese

Who Should Purchase:

IT professionals, career-changers, college students, recent graduates and executives seeking foundational knowledge in cybersecurity. ISC2 is offering free Certified in Cybersecurity (CC) Online Self-Paced Training and an exam to 1 million people as part of our pledge to help close the cybersecurity workforce gap and diversify those working in the field. To

meet every learner’s needs, we’re also offering two CC Training Bundles with special extras. Learn more.

What to Expect:

A comprehensive review of the Official ISC2 Training Course content related to the ISC2 Certified in Cybersecurity exam. Learning supports include:

- Chapter overviews, objectives and summaries
- Informative graphics
- Key terms and definitions
- Chapter quizzes and answer keys

CPE Credits

None

Access Period:

365 days from inital access

This eTextbook covers the following:

Chapter 1: Security Principles

- • 1: Understand the Security Concepts of Information Assurance
- • 2: Understand the Risk Management Process
- • 3: Understand Security Controls
- • 4: Understand Governance Elements and Processes
- • 5: Understand ISC2 Code of Ethics

Chapter 2: Incident Response, Business Continuity and Disaster Recovery Concepts

- • 1: Understand Incident Response
- • 2: Understand Business Continuity
- • 3: Understand Disaster Recovery

Chapter 3: Access Controls Concepts

- • 1: Understand Access Control Concepts
- • 2: Understand Physical Access Controls

- • 3: Understand Logical Access Controls

Chapter 4: Network Security

- • 1: Understand Computer Networking
- • 2: Understand Network (Cyber) Threats and Attacks
- • 3: Understand Network Security Infrastructure

Chapter 5: Security Operations

- • 1: Understand Data Security
- • 2: Understand System Hardening
- • 3: Understand Best Practice Security Policies
- • 4: Understand Security Awareness Training

Technology Requirements:

The CC eTextbook uses VitalSource eReader, which will allow you to view materials on multiple devices and platforms, online and offline.

The following may be among system requirements to access your eTextbook.

- A stable and continuous internet connection.

Hardware Specifications

- • Processor 2 GHz +
- • RAM 4 GB +
- • Monitor minimum resolution (1024 x 768)
- • Video Card
- • Keyboard and Mouse or other assistive technology

Supported Operating Systems

- • Macintosh OS X 10.10 to present
- • Windows 10 to present

Supported Browsers

- • Google Chrome
- • Microsoft Edge
- • Mozilla Firefox

Application Software

- VitalSource eReader

Certification **Exam Outline**

- Effective Date: November 15, 2022

About CISSP-ISSMP

The Information Systems Security Management Professional (ISSMP) is a CISSP who specializes in establishing, presenting and governing information security programs and demonstrates management and leadership skills. CISSP-ISSMPs direct the alignment of security programs with the organization's mission, goals and strategies in order to meet enterprise financial and operational requirements in support of its desired risk position.

The broad spectrum of topics included in the CISSP-ISSMP Common Body of Knowledge (CBK®) ensure its relevancy across all disciplines in the field of information security management. Successful candidates are competent in the following six domains:

- • Leadership and Business Management
- • Systems Lifecycle Management
- • Risk Management
- • Threat Intelligence and Incident Management
- • Contingency Management
- • Law, Ethics and Security Compliance Management

Experience Requirements

Candidates must be a CISSP in good standing and have two years cumulative paid work experience in one or more of the six domains of the CISSP-ISSMP CBK. You can learn more about CISSP-ISSMP experience requirements and how to account for part-time work and internships at www.isc2.org/Certifications/CISSP-Concentrations#steps-to-certification.

Accreditation

CISSP-ISSMP is in compliance with the stringent requirements of ANSI/ISO/IEC Standard 17024.

Job Task Analysis (JTA)

(ISC)² has an obligation to its membership to maintain the relevancy of the CISSP-ISSMP. Conducted at regular intervals, the Job Task Analysis (JTA) is a methodical and critical process of determining the tasks that are performed by security professionals who are engaged in the profession defined by the CISSP-ISSMP. The

- results of the JTA are used to update the examination. This process ensures that candidates are tested on the
- topic areas relevant to the roles and responsibilities of today’s practicing information security professionals.
-

CISSP-ISSMP Examination Information

CISSP-ISSMP Examination Weights

- **Length of exam**
- **Number of items**
- **Item format**
- **Passing grade**
- **Exam availability**
- **Testing center**
- 3 hours
- 125
- Multiple choice
- 700 out of 1000 points
- English
- Pearson VUE Testing Center
-
-
- 1. Leadership and Business Management
- 20%
- 2. Systems Lifecycle Management
- 18%
- 3. Risk Management
- 19%
- 4. Threat Intelligence and Incident Management
- 17%
- 5. Contingency Management
- 15%
- 6. Law, Ethics and Security Compliance Management
- 11%
- **Total: 100%**
-

Domain 1:

Leadership and Business Management

- 1.1 Establish security’s role in organizational culture, vision and mission

- » Define information security program vision and mission
- » Align security with organizational goals, objectives and values
- » Define security’s relationship to the overall business processes
- » Define the relationship between organizational culture and security
- 1.2 Align security program with organizational governance
- » Identify and navigate organizational governance structure
- » Validate roles of key stakeholders
- » Validate sources and boundaries of authorization
- » Advocate and obtain organizational support for security initiatives
- 1.3 Define and implement information security strategies
- » Identify security requirements from business initiatives
- » Evaluate capacity and capability to implement security strategies
- » Manage implementation of security strategies
- » Review and maintain security strategies
- » Prescribe security architecture and engineering theories, concepts and methods
- 1.4 Define and maintain security policy framework Determine applicable external standards
- » Determine applicable external standards
- » Determine data classification and protection requirements
- » Establish internal policies
- » Advocate and obtain organizational support for policies
- » Develop procedures, standards, guidelines and baselines
- » Ensure periodic review of security policy framework
-
- » Define roles and responsibilities
- » Determine and manage team accountability
- » Build cross-functional relationships
- » Resolve conflicts between security and
- other stakeholders
- » Identify communication bottlenecks
- and barriers
- » Integrate security controls into human
- resources processes
- » Evaluate service management agreements
- (e.g., risk, financial)
- » Govern managed services
- (e.g., infrastructure, cloud services)
- » Manage impact of organizational change (e.g.,
- mergers and acquisitions, outsourcing)
- » Ensure that appropriate regulatory compliance
- statements and requirements are included in
- contractual agreements
- » Monitor and enforce compliance with
- contractual agreements
- 1.5 Manage security requirements in contracts and agreements
- 1.6 Manage security awareness and training programs
- » Promote security programs to key stakeholders
- » Identify needs and implement training programs by target segment
- » Monitor and report on effectiveness of security awareness and training programs
- 1.7 Define, measure and report security metrics
- » Identify Key Performance Indicators (KPI)
- » Associate Key Performance Indicators (KPI) to the risk posture of the organization
- » Use metrics to drive security program development and operations
- 1.8 Prepare, obtain and administer security budget

- » Prepare and secure annual budget
- » Adjust budget based on evolving risks and threat landscape
- » Manage and report financial responsibilities
- 1.9 Manage security programs
- 1.10 Apply product development and project management principles
- » Incorporate security into project lifecycle
- » Identify and apply appropriate project management methodology
- » Analyze project time, scope and cost relationship
-
- 2.1 Manage integration of security into Systems Development Life Cycle (SDLC)
- » Integrate information security gates (decision points) and requirements into lifecycle
- » Implement security controls into system lifecycle
- » Oversee security configuration management (CM) processes
- 2.2 Integrate new business initiatives and emerging technologies into the
- security architecture
- » Integrate security into new business initiatives and emerging technologies
- » Address impact of new business initiatives on security posture
- 2.3 Define and oversee comprehensive vulnerability management programs
- (e.g., vulnerability scanning, penetration testing, threat analysis)
- » Identify, classify and prioritize assets, systems and services based on criticality to business
- » Prioritize threats and vulnerabilities
- » Manage security testing
- » Manage mitigation and/or remediation of vulnerabilities based on risk
- 2.4 Manage security aspects of change control
- » Integrate security requirements with change control process
- » Identify and coordinate with the stakeholders
- » Manage documentation and tracking
- » Ensure policy compliance (e.g., continuous monitoring)
- **Domain 2:**
- **Systems Lifecycle Management**
-
- **Domain 3:**
- **Risk Management**
- 3.1 Develop and manage a risk management program
- » Identify risk management program objectives
- » Communicate and agree on risk management objectives with risk owners and other stakeholders
- » Determine scope of organizational risk program
- » Identify organizational security risk tolerance/appetite
- » Obtain and verify organizational asset inventory
- » Analyze organizational risks

- » Determine countermeasures, compensating and mitigating controls
- » Perform cost-benefit analysis (CBA) of risk treatment options
- 3.2 Conduct risk assessments
- » Identify risk factors
- 3.3 Manage security risks within the supply chain (e.g., supplier, vendor, third-party risk)
- » Identify supply chain security risk requirements
- » Integrate supply chain security risks into organizational risk management
- » Validate security risk control within the supply chain
- » Monitor and review the supply chain security risks
-
- 4.1 Establish and maintain threat intelligence program
- » Aggregate threat data from multiple threat intelligence sources
- » Conduct baseline analysis of network traffic, data and user behavior
- » Detect and analyze anomalous behavior patterns for potential concerns
- » Conduct threat modeling
- » Identify and categorize an attack
- » Correlate related security event and threat data
- » Create actionable alerting to appropriate resources
- 4.2 Establish and maintain incident handling and investigation program
- » Develop program documentation
- » Establish incident response case management process
- » Establish incident response team
- » Apply incident management methodologies
- » Establish and maintain incident handling process
- » Establish and maintain investigation process
- » Quantify and report financial and operational impact of incidents and investigations to stakeholders
- » Conduct root cause analysis (RCA)
- # Domain 4:
- # Threat Intelligence and Incident
- # Management
-
- 5.1 Facilitate development of contingency plans
- » Identify and analyze factors related to the Continuity of Operations Plan (COOP)
- » Identify and analyze factors related to the business continuity plan (BCP) (e.g., time, resources, verification)
- » Identify and analyze factors related to the disaster recovery plan (DRP) (e.g., time, resources, verification)
- » Coordinate contingency management plans with key stakeholders
- » Define internal and external crisis communications plans
- » Define and communicate contingency roles and responsibilities
- » Identify and analyze contingency impact on business processes and priorities
- » Manage third-party contingency dependencies
- » Prepare security management succession plan
- 5.2 Develop recovery strategies
- » Identify and analyze alternatives

- » Recommend and coordinate recovery strategies
- » Assign recovery roles and responsibilities
- 5.3 Maintain contingency plan, Continuity of Operations Plan (COOP), business continuity
- plan (BCP) and disaster recovery plan (DRP)
- » Plan testing, evaluation and modification
- » Determine survivability and resiliency capabilities
- » Manage plan update process
- 5.4 Manage disaster response and recovery process
- » Declare disaster
- » Implement plan
- » Restore normal operations
- » Gather lessons learned
- » Update plan based on lessons learned
- # Domain 5:
- # Contingency Management
-
-
- 6.1 Identify the impact of laws and regulations that relate to information security
- 6.2 Adhere to the (ISC)
- [2]
- Code of Ethics as related to management issues
- 6.3 Validate compliance in accordance with applicable laws, regulations and industry
- best practices
- 6.4 Coordinate with auditors and regulators in support of the internal and external
- audit processes
- 6.5 Document and manage compliance exceptions
- » Identify and document compensating controls and workarounds
- » Report and obtain authorized approval of risk waiver
- # Domain 6:
- # Law, Ethics and Security Compliance
- # Management
- » Identify applicable privacy laws
- » Identify legal jurisdictions the organization and
- users operate within (e.g., trans-border data flow)
- » Identify export laws
- » Identify intellectual property (IP) laws
- » Identify applicable industry regulations
- » Identify and advise on non-compliance risks
- » Inform and advise senior management

- » Evaluate and select compliance framework(s)
- » Implement the compliance framework(s)
- » Define and monitor compliance metrics
- » Plan
- » Schedule
- » Coordinate audit activities
- » Evaluate and validate findings
- » Formulate response
- » Validate implemented mitigation and
- remediation actions
-

Additional Examination Information

Supplementary References

- Candidates are encouraged to supplement their education and experience by reviewing
- relevant resources that pertain to the CBK and identifying areas of study that may need
- additional attention.
- View the full list of supplementary references at www.isc2.org/certifications/References.

Examination Policies and Procedures

- (ISC)₂ recommends that CISSP-ISSMP candidates review exam policies and procedures
- prior to registering for the examination. Read the comprehensive breakdown of this
- important information at www.isc2.org/Exams/Before-Your-Exam.

Legal Info

- For any questions related to (ISC)
- ²
- ’s legal policies, please contact the (ISC)₂ Legal
- Department at legal@isc2.org.

Any Questions?

- (ISC)₂ Americas
- Tel: +1.866.331.ISC2 (4722)
- Email: info@isc2.org
- (ISC)₂ Asia-Pacific
- Tel: +(852) 28506951
- Email: isc2asia@isc2.org
- (ISC)₂ EMEA
- Tel: +44 (0)203 300 1625
- Email: info-emea@isc2.org
-

v222Register for exam

Effective Date: August 29, 2022

Certified in Cybersecurity Certification Exam Outline

View and download the latest PDF version of the Certified in Cybersecurity Exam Outline in the following languages:

CC - English | CC - Chinese | CC - Japanese | CC - German | CC - Spanish

About Certified in Cybersecurity

ISC2 developed the Certified in Cybersecurity (CC) credential for newcomers to the field, to recognize the growing trend of people entering the cybersecurity workforce without direct IT experience. Getting Certified in Cybersecurity provides employers with the confidence that you have a solid grasp of the right technical concepts, and a demonstrated aptitude to learn on the job. As an ISC2 certification, those who hold the CC are backed by the world's largest network of certified cybersecurity professionals helping them continue their professional development and earn new achievements and qualifications throughout their career.

The topics on the CC exam include:

- Security Principles
- Incident Response, Business Continuity (BC) and Disaster Recovery (DR) Concepts
- Access Controls Concepts
- Network Security
- Security Operations

Certified in Cybersecurity Examination Information

Length of exam	2 hours
Number of items	100
Item format	Multiple choice
Passing grade	700 out of 1000 points
Exam language availability	English, Chinese, Japanese, German, Spanish
Testing center	Pearson VUE Testing Center

Certified in Cybersecurity Examination Weights

Domains	Average Weight
1. Security Principles	26%
2. Business Continuity (BC), Disaster Recovery (DR) & Incident Response Concepts	10%
3. Access Controls Concepts	22%
4. Network Security	24%
5. Security Operations	18%
Total	**100%**

Boost your chances of passing the exam with the CC eTextbook
Build your fundamental cybersecurity knowledge and prepare for the exam.

Get eTextbook

Domains

1.1 - Understand the security concepts of information assurance

- Confidentiality
- Integrity
- Availability
- Authentication (e.g., methods of authentication, multi-factor authentication (MFA))
- Non-repudiation
- Privacy

1.2 - Understand the risk management process

- Risk management (e.g., risk priorities, risk tolerance)
- Risk identification, assessment and treatment

1.3 - Understand security controls

- Technical controls
- Administrative controls
- Physical controls

1.4 - Understand ISC2 Code of Ethics

- Professional code of conduct

1.5 - Understand governance processes

- Policies
- Procedures
- Standards
- Regulations and laws

2.1 - Understand business continuity (BC)

- Purpose
- Importance
- Components

2.2 - Understand disaster recovery (DR)

- Purpose
- Importance
- Components

2.3 - Understand incident response

- Purpose
- Importance
- Components

3.1 - Understand physical access controls

- Physical security controls (e.g., badge systems, gate entry, environmental design)
- Monitoring (e.g., security guards, closed-circuit television (CCTV), alarm systems, logs)
- Authorized versus unauthorized personnel

3.2 - Understand logical access controls

- Principle of least privilege
- Segregation of duties
- Discretionary access control (DAC)
- Mandatory access control (MAC)
- Role-based access control (RBAC)

4.1 - Understand computer networking

- Networks (e.g., Open Systems Interconnection (OSI) model, Transmission Control Protocol/Internet Protocol (TCP/IP) model, Internet Protocol version 4 (IPv4), Internet Protocol version 6 (IPv6), WiFi)
- Ports
- Applications

4.2 - Understand network threats and attacks

- Types of threats (e.g., distributed denial-of-service (DDoS), virus, worm, Trojan, man-in-the-middle (MITM), side-channel)
- Identification (e.g., intrusion detection system (IDS), host-based intrusion detection system (HIDS), network intrusion detection system (NIDS))
- Prevention (e.g., antivirus, scans, firewalls, intrusion prevention system (IPS))

4.3 - Understand network security infrastructure

- On-premises (e.g., power, data center/closets, Heating, Ventilation, and Air Conditioning (HVAC), environmental, fire suppression, redundancy, memorandum of understanding (MOU)/memorandum of agreement (MOA))
- Design (e.g., network segmentation (demilitarized zone (DMZ), virtual local area network (VLAN), virtual private network (VPN), micro-segmentation), defense in depth, Network Access Control (NAC) (segmentation for embedded systems, Internet of Things (IoT))
- Cloud (e.g., service-level agreement (SLA), managed service provider (MSP), Software as a Service (SaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS), hybrid)

5.1 - Understand data security

- Encryption (e.g., symmetric, asymmetric, hashing)
- Data handling (e.g., destruction, retention, classification, labeling)
- Logging and monitoring security events

5.2 - Understand system hardening

- Configuration management (e.g., baselines, updates, patches)

5.3 - Understand best practice security policies

- Data handling policy
- Password policy
- Acceptable Use Policy (AUP)
- Bring your own device (BYOD) policy
- Change management policy (e.g., documentation, approval, rollback)
- Privacy policy

5.4 - Understand security awareness training

- Purpose/concepts (e.g., social engineering, password protection)
- Importance

A safe and secure cyber world

The Center for Cyber Safety & EducationISC2 CareersCommunityBlog

Frequently Asked QuestionsContact UsPolicies and Procedures

ISC2 Authorized China AgencyISC2 Japan

© Copyright 1996-2025. ISC2, Inc. All Rights Reserved.

All contents of this site constitute the property of ISC2, Inc. and may not be copied, reproduced or distributed without prior written permission. ISC2, CISSP, SSCP, CCSP, CGRC, CSSLP, HCISPP, ISSAP, ISSEP, ISSMP, CC, and CBK are registered marks of ISC2, Inc.

Sitemap

Access Period:

365 days from inital access

This eTextbook covers the following:

Chapter 1: Security Principles

- • 1: Understand the Security Concepts of Information Assurance
- • 2: Understand the Risk Management Process
- • 3: Understand Security Controls
- • 4: Understand Governance Elements and Processes
- • 5: Understand ISC2 Code of Ethics

Chapter 2: Incident Response, Business Continuity and Disaster Recovery Concepts

- • 1: Understand Incident Response
- • 2: Understand Business Continuity
- • 3: Understand Disaster Recovery

Chapter 3: Access Controls Concepts

- • 1: Understand Access Control Concepts
- • 2: Understand Physical Access Controls
- • 3: Understand Logical Access Controls

Chapter 4: Network Security

- • 1: Understand Computer Networking
- • 2: Understand Network (Cyber) Threats and Attacks
- • 3: Understand Network Security Infrastructure

Chapter 5: Security Operations

- • 1: Understand Data Security
- • 2: Understand System Hardening
- • 3: Understand Best Practice Security Policies
- • 4: Understand Security Awareness Training

Technology Requirements:

The CC eTextbook uses VitalSource eReader, which will allow you to view materials on multiple devices and platforms, online and offline.

The following may be among system requirements to access your eTextbook.

- A stable and continuous internet connection.

Hardware Specifications

- • Processor 2 GHz +
- • RAM 4 GB +
- • Monitor minimum resolution (1024 x 768)
- • Video Card
- • Keyboard and Mouse or other assistive technology

Supported Operating Systems

- • Macintosh OS X 10.10 to present
- • Windows 10 to present

Supported Browsers

- • Google Chrome
- • Microsoft Edge
- • Mozilla Firefox

Application Software

- VitalSource eReader

Online Training

Continuing education

Stay relevant and on top of the latest trends. Leverage our online courses to gain interactive, engaging and timely learning experiences throughout your career. Each course is designed with input from leading industry experts and based on proven learning techniques to maximize your time and content retention.

The self-paced learning format delivers modular content combined with interactive activities involving videos, labs, case studies, quizzes, etc. Learn at your own pace on your own time and earn valuable Continuing Professionals Education (CPE) credits towards your ISC2; certifications. (Note: Credits may be eligible for continuing education credits for non-ISC2 certifications. Please review the requirements established by the credentialing organization for eligibility.)

ISC2; Members and Associates have free unlimited access to courses that are denoted as ‘Free for Members’ by logging in above and clicking the ‘My Courses’ menu item.

For individual course purchases learners will have access to the course content for 180 days from the time of purchase. Please note, we do not offer time extensions for you to complete this training course.

For All Access and Express Learning Bundle purchases learners will have access to the course content for 365 day from the time of purchase. Please see course description for pricing and CPEs.

1 2 3 > >|

Certification

Defining the Boundaries of Zero Trust

This learning experience invites you to review the set of guiding principles for workflow, system design, and operations that create a zero trust architecture. (2.0 CPE)

-

Software Inventory and SBOM

This course invites you to expand your knowledge of how Software Bill of Materials (SBOM) can help cybersecurity professionals effectively mitigate vulnerabilities and ensure compliance.

Software Inventory and SBOM

This course invites you to expand your knowledge of how Software Bill of Materials (SBOM) can help cybersecurity professionals effectively mitigate vulnerabilities and ensure compliance.

Working in the Cloud

This course invites you to learn about the range of challenges security professionals face as they work to utilize, optimize and secure critical assets in the cloud.

Working in the Cloud

This course invites you to learn about the range of challenges security professionals face as they work to utilize, optimize and secure critical assets in the cloud.

Moving to the Cloud

This course invites you to learn about the strategic and security considerations necessary to transition an organization to cloud computing in alignment with business needs.

Moving to the Cloud

This course invites you to learn about the strategic and security considerations necessary to transition an organization to cloud computing in alignment with business needs.

Cloud Basics

This course invites you to learn about essential cloud concepts and principles, including key drivers for use, essential characteristics, and service and deployment models within cloud architectures.

Cloud Basics

This course invites you to learn about essential cloud concepts and principles, including key drivers for use, essential characteristics, and service and deployment models within cloud architectures.

Building Your Personal Brand and Digital Presence

This course invites you to review the essentials of building a personal brand and digital presence that reflects your core values, unique strengths and professional aspirations.

Resume/CV/Portfolio Building and Management

This course invites you to review the critical components of documents such as resumes, CVs and portfolios that showcase your unique strengths and value to potential employers.

Identifying and Building Your Network

This course invites you to review the critical role networking plays in professional development by unlocking new opportunities, facilitating knowledge sharing and supporting long-term career success.

Identifying Your Cyber Path and Industry

This course invites you to explore a wide range of opportunities in cybersecurity and plan a skill development path toward a successful career.

Nailing the Interview Process

This course invites you to review actionable strategies and skills to excel in any interview scenario and stand out from the competition.

Managing the Offer and Negotiation Process

This course invites you to review the essential knowledge and skills to navigate job offers, negotiate confidently and transition smoothly into new roles.

Managing the Offer and Negotiation Process

This course invites you to review the essential knowledge and skills to navigate job offers, negotiate confidently and transition smoothly into new roles.

Nailing the Interview Process

This course invites you to review actionable strategies and skills to excel in any interview scenario and stand out from the competition.

Identifying Your Cyber Path and Industry

This course invites you to explore a wide range of opportunities in cybersecurity and plan a skill development path toward a successful career.

Identifying and Building Your Network

This course invites you to review the critical role networking plays in professional development by unlocking new opportunities, facilitating knowledge sharing and supporting long-term career success.

Resume/CV/Portfolio Building and Management

This course invites you to review the critical components of documents such as resumes, CVs and portfolios that showcase your unique strengths and value to potential employers.

Go to the main content

Privacy and Cookies This website stores cookies on your computer which help us make the website work better for you.

Collapsed, toggle side navigation to expand
Edit my profile Tshingombe Tshingombe Tshitadi
ISC2 ID: 1907033

- Dashboard
- My Profile
-
-
- Sign out

Find an exam

Find an Exam:

CC	Certified in Cybersecurity (CC)
CCSP	Certified Cloud Security Professional (CCSP)
CGRC	Certified in Governance Risk and Compliance
CISSP	Certified Information Systems Security Professional
CSSLP	Certified Secure Software Lifecycle Professional
ISSAP	Information Systems Security Architecture Professional
ISSEP	Information Systems Security Engineering Professional
ISSMP	Information Systems Security Management Professional
SSCP	Systems Security Certified Practitioner

- Terms
- Privacy
- Contact

Copyright 1996-2025 Pearson Education Inc. or its affiliate(s). All rights reserved.

-

Certification is Just the Beginning

Achieving your certification isn’t the end of your journey. ISC2 members and associates are part of a

Register for exam

Effective Date: September 2024

SSCP Certification Exam Outline

View and download the latest PDF version of the SSCP Certification Exam Outline in the following languages:

SSCP - English | SSCP - Japanese | SSCP - Spanish

About SSCP

The Systems Security Certified Practitioner (SSCP) is the ideal certification for those with proven technical skills and practical, hands-on security knowledge in operational IT roles. It provides confirmation of a practitioner's ability to implement, monitor and administer IT infrastructure in accordance with information security policies and procedures that ensure data confidentiality, integrity and availability.

The broad spectrum of topics included in the SSCP Common Body of Knowledge (CBK) ensure its relevancy across all disciplines in the field of information security. Successful candidates are competent in the following domains:

- Security Concepts and Practices
- Access Controls
- Risk Identification, Monitoring, and Analysis
- Incident Response and Recovery
- Cryptography
- Network and Communications Security
- Systems and Application Security

Experience Requirements

Candidates must have a minimum of one year cumulative work experience in one or more of the domains of the SSCP CBK. A one year prerequisite pathway will be granted for candidates who received a degree (bachelors or masters) in a cybersecurity program.

A candidate that doesn't have the required experience to become an SSCP may become an Associate of ISC2 by successfully passing the SSCP examination. The Associate of ISC2 will then have two years to earn the one year required experience. You can learn more about SSCP experience requirements and how to account for part-time work and internships at www.isc2.org/Certifications/SSCP/SSCP-Experience-Requirements.

Accreditation

SSCP is in compliance with the stringent requirements of ANSI/ISO/IEC Standard 17024.

Job Task Analysis (JTA)

ISC2 has an obligation to its membership to maintain the relevancy of the SSCP. Conducted at regular intervals, the Job Task Analysis (JTA) is a

methodical and critical process of determining the tasks that are performed by security professionals who are engaged in the profession defined by the SSCP. The results of the JTA are used to update the examination. This process ensures that candidates are tested on the topic areas relevant to the roles and responsibilities of today's practicing information security professionals.

SSCP Examination Information

Length of exam	3 hours
Number of items	125
Item format	Multiple choice
Passing grade	700 out of 1000 points
Language availability	English, Japanese and Spanish
Testing center	Pearson VUE Testing Center

SSCP Examination Weights

Domains	Average Weight
1. Security Concepts and Practices	16%
2. Access Controls	15%
3. Risk Identification, Monitoring and Analysis	15%
4. Incident Response and Recovery	14%
5. Cryptography	9%
6. Network and Communications Security	16%
7. Systems and Application Security	15%
Total	**100%**

Domains

1.1 - Comply with codes of ethics

- ISC2 Code of Ethics
- Organizational code of ethics

1.2 - Understand security concepts

- Confidentiality
- Integrity
- Availability
- Accountability
- Non-repudiation
- Least privilege
- Segregation of duties (SoD)

1.3 - Identify and implement security controls

- Technical controls (e.g., firewalls, intrusion detection systems (IDS), access control list (ACL)
- Physical controls (e.g., mantraps, cameras, locks)
- Administrative controls (e.g., security policies, standards, procedures, baselines)
- Assessing compliance requirements
- Periodic audit and review

1.4 - Document and maintain functional security controls

- Deterrent controls
- Preventative controls
- Detective controls
- Corrective controls
- Compensating controls

1.5 - Support and implement asset management lifecycle (i.e., hardware, software, and data)

- Process, planning, design and initiation
- Development /Acquisition (e.g., DevSecOps, testing)
- Inventory and licensing (e.g., open source, closed-source)
- Implementation/Assessment
- Operation/Maintenance/End of Life (EOL)
- Archival and retention requirements
- Disposal and destruction

1.6 - Support and/or implement change management lifecycle

- Change management (e.g., roles, responsibilities, processes, communications, audit)
- Security impact analysis
- Configuration management (CM)

1.7 - Support and/or implement security awareness and training (e.g., social engineering/phishing/tabletop exercises/awareness communications)

1.8 - Collaborate with physical security operations (e.g., data center/facility assessment, badging and visitor management, personal device restrictions)

2.1 - Implement and maintain authentication methods

- Single/Multi-factor authentication (MFA)
- Single sign-on (SSO) (e.g., Active Directory Federation Services (ADFS), OpenID Connect)
- Device authentication (e.g., certificate, Media Access Control (MAC) address, Trusted Platform Module (TPM))
- Federated access (e.g., Open Authorization 2 (OAuth2), Security Assertion Markup Language (SAML))

2.2 - Understand and support internetwork trust architectures

- Trust relationships (e.g., 1-way, 2-way, transitive, zero)
- Internet, intranet, extranet, and demilitarized zone (DMZ)
- Third-party connections (e.g., application programming interface (API), app extensions, middleware)

2.3 - Support and/or implement the identity management lifecycle

- Authorization
- Proofing
- Provisioning/De-provisioning
- Monitoring, Reporting, and Maintenance (e.g., role changes, new security standards)
- Entitlement (e.g., inherited rights, resources)
- Identity and access management (IAM) systems

2.4 - Understand and administer access controls

- Mandatory
- Discretionary
- Role-based (e.g., subject-based, object-based, Privileged Access Management (PAM))
- Rule-based
- Attribute-based

3.1 - Understand risk management

- Risk visibility and reporting (e.g., risk register, sharing threat intelligence, indicators of Compromise (IOC), Common Vulnerability Scoring System (CVSS), socialization, MITRE/ATT&CK model)
- Risk management concepts (e.g., impact assessments, threat modeling, scope)
- Risk management frameworks
- Risk tolerance (e.g., appetite, risk quantification)
- Risk treatment (e.g., accept, transfer, mitigate, avoid)

3.2 - Understand legal and regulatory concerns (e.g., jurisdiction, limitations, privacy)

3.3 - Perform security assessments and vulnerability management activities

- Risk management frameworks implementation
- Security testing

- Risk review (e.g., internal, supplier, architecture)
- Vulnerability management lifecycle (e.g., scanning, reporting, analysis, remediation)

3.4 - Operate and monitor security platforms (e.g., continuous monitoring)

- Source systems (e.g., applications, security appliances, network devices, hosts)
- Events of interest (e.g., errors, omissions, anomalies, unauthorized changes, compliance violations, policy failures)
- Log management (e.g., policy, integrity, preservation, architectures, configuration, aggregation, tuning)
- Security information and event management (SIEM) (e.g., real-time monitoring, analysis, tracking, audit)

3.5 - Analyze monitoring results

- Security baselines and anomalies (e.g., correlation, noise reduction)
- Visualizations, metrics, and trends (e.g., notifications, dashboards, timelines)
- Event data analysis
- Document and communicate findings (e.g., escalation)

4.1 - Understand and support incident response lifecycle (e.g., National Institute of Standards and Technology (NIST), International Organization for Standardization (ISO))

- Preparation (e.g., defining roles, training programs)
- Detection, analysis, and escalation (e.g., incident communication, public relations)
- Containment
- Eradication
- Recovery (e.g., incident documentation)
- Post incident activities (e.g., lessons learned, new countermeasures, continuous improvement)

4.2 - Understand and support forensic investigations

- Legal (e.g., civil, criminal, administrative) and ethical principles
- Evidence handling (e.g., first responder, triage, chain of custody, preservation of scene)
- Reporting of analysis
- Organization Security Policy Compliance

4.3 - Understand and support business continuity plan (BCP) and disaster recovery plan (DRP)

- Emergency response plans and procedures (e.g., information system contingency, pandemic, natural disaster, crisis management)
- Interim or alternate processing strategies
- Restoration planning (e.g., Restore Time Objective (RTO), Restore Point Objectives (RPO), Maximum Tolerable Downtime (MTD))
- Backup and redundancy implementation
- Testing and drills (e.g., playbook, tabletop, disaster recovery exercises, scheduling)

5.1 - Understand reasons and requirements for cryptography

- Confidentiality
- Integrity and authenticity
- Data sensitivity (e.g., personally identifiable information (PII), intellectual property (IP), protected health information (PHI))
- Regulatory and industry best practice (e.g., Payment Card Industry Data Security Standards (PCI-DSS), International Organization for Standardization (ISO))
- Cryptography entropy (e.g., quantum cryptography, quantum key distribution)

5.2 - Apply cryptography concepts

- Hashing
- Salting
- Symmetric/Asymmetric encryption/Elliptic curve cryptography (ECC)
- Non-repudiation (e.g., digital signatures/certificates, Hash-based Message Authentication Code (HMAC), audit trails)
- Strength of encryption algorithms and keys (e.g., Advanced Encryption Standards (AES), Rivest-Shamir-Adleman (RSA)
- Cryptographic attacks and cryptanalysis

5.3 - Understand and implement secure protocols

- Services and protocols
- Common use cases (e.g., credit card processing, file transfer, web client, virtual private network (VPN), transmission of PII data)
- Limitations and vulnerabilities

5.4 - Understand public key infrastructure (PKI)

- Fundamental key management concepts (e.g., storage, rotation, composition, generation, destruction, exchange, revocation, escrow)
- Web of Trust (WOT) (e.g., Pretty Good Privacy (PGP), GNU Privacy Guard (GPG), blockchain)

6.1 - Understand and apply fundamental concepts of networking

- Open Systems Interconnection (OSI) and Transmission Control Protocol/Internet Protocol (TCP/IP) models
- Network topologies
- Network relationships (e.g., peer-to-peer (P2P), client server)
- Transmission media types (e.g., wired, wireless)
- Software-defined networking (SDN) (e.g., Software-Defined Wide Area Network (SD-WAN), network virtualization, automation)
- Commonly used ports and protocols

6.2 - Understand network attacks (e.g., distributed denial of service (DDoS), man-in-the-middle (MITM), Domain Name System (DNS) cache poisoning)

- Countermeasures (e.g., content delivery networks (CDN), firewalls, network access controls, intrusion detection and prevention systems (IDPS))

6.3 - Manage network access controls

- Network access controls, standards and protocols (e.g., Institute of Electrical and Electronics Engineers (IEEE) 802.1X, Remote Authentication Dial-In User Service (RADIUS), Terminal Access Controller Access-Control System Plus (TACACS+))
- Remote access operation and configuration (e.g., thin client, virtual private network (VPN), virtual desktop infrastructure)

6.4 - Manage network security

- Logical and physical placement of network devices (e.g., inline, passive, virtual)
- Segmentation (e.g., physical/logical, data/control plane, virtual local area network (VLAN), access control list (ACL), firewall zones, micro-segmentation)
- Secure device management

6.5 - Operate and configure network-based security appliances and services

- Firewalls and proxies (e.g., filtering methods, web application firewall (WAF), cloud access security broker (CASB))
- Network intrusion detection/prevention systems
- Routers and switches
- Traffic-shaping devices (e.g., wide area network (WAN) optimization, load balancing)
- Network Access Control (NAC)
- Data Loss Prevention (DLP)
- Unified Threat Management (UTM)

6.6 - Secure wireless communications

- Technologies (e.g., cellular network, Wi-Fi, Bluetooth, Near-Field Communication (NFC))
- Authentication and encryption protocols (e.g., Wi-Fi Protected Access (WPA), Extensible Authentication Protocol (EAP), Wi-Fi Protected Access 2 (WPA2), Wi-Fi Protected Access 3 (WPA3))

6.7 Secure and monitor Internet of Things (IoT) (e.g., configuration, network isolation, firmware updates, End of Life (EOL) management)

7.1 - Identify and analyze malicious code and activity

- Malware (e.g., rootkits, spyware, scareware, ransomware, trojans, virus, worms, trapdoors, backdoors, fileless, app/code/operatin3 system (OS)/mobile code vulnerabilities)
- Malware countermeasures (e.g., scanners, anti-malware, containment and remediation, software security)
- Types of malicious activity (e.g., insider threat, data theft, distributed denial of service (DDoS), botnet, zero-day exploits, web-based attacks, advanced persistent threat (APT))
- Malicious activity countermeasures (e.g., user awareness/training, system hardening, patching, isolation, data loss prevention (DLP))
- Social engineering methods (e.g., SPAM email, phishing/smishing/vishing, impersonation, scarcity, whaling)
- Behavior analytics (e.g., machine learning, Artificial Intelligence (AI), data analytics)

7.2 - Implement and operate endpoint device security

- Host-based intrusion prevention system (HIPS)
- Host-based intrusion detection system (HIDS)
- Host-based firewalls
- Application white listing
- Endpoint encryption (e.g., full disk encryption)
- Trusted Platform Module (TPM) (e.g., hardware security module management)
- Secure browsing (e.g., digital certificates)
- Endpoint detection and response (EDR)

7.3 - Administer and manage mobile devices

- Provisioning techniques (e.g., corporate owned, personally enabled (COPE), Bring Your Own Device (BYOD), Mobile Device Management (MDM))
- Containerization
- Encryption
- Mobile application management

7.4 - Understand and configure cloud security

- Deployment models (e.g., public, private, hybrid, community)

- Service models (e.g., Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS))
- Virtualization (e.g., hypervisor, Virtual Private Cloud (VPC))
- Legal and regulatory concerns (e.g., privacy, surveillance, data ownership, jurisdiction, eDiscovery, shadow information technology (IT))
- Data storage, processing, and transmission (e.g., archiving, backup, recovery, resilience)
- Third-party/Outsourcing requirements (e.g., service-level agreement (SLA), data portability/ privacy/destruction/auditing)
- Shared responsibility model

7.5 - Operate and maintain secure virtual environments

- Hypervisor (i.e., Type 1 (e.g., bare metal), Type 2 (e.g., software))
- Virtual appliances
- Containers
- Continuity and resilience
- Storage management (e.g., data domain)
- Threats, attacks, and countermeasures (e.g., brute-force attack, virtual machine escape, threat hunting)

Additional Examination Information

Supplementary References

Candidates are encouraged to supplement their education and experience by reviewing relevant resources that pertain to the CBK and identifying areas of study that may need additional attention.

View the full list of supplementary references at www.isc2.org/certifications/References.

Examination Policies and Procedures

ISC2 recommends that SSCP candidates review exam policies and procedures prior to registering for the examination. Read the comprehensive breakdown of this important information at www.isc2.org/Register-for-Exam.

The Center for Cyber Safety & EducationISC2 CareersCommunityBlog
Frequently Asked QuestionsContact UsPolicies and Procedures
ISC2 Authorized China AgencyISC2 Japan

© Copyright 1996-2025. ISC2, Inc. All Rights Reserved.

All contents of this site constitute the property of ISC2, Inc. and may not be copied, reproduced or distributed without prior written permission. ISC2, CISSP, SSCP, CCSP, CGRC, CSSLP, HCISPP, ISSAP, ISSEP, ISSMP, CC, and CBK are registered marks of ISC2, Inc.

Sitemap

SSCP Exam Outline

Agree to ISC2 policies

back to previous step
CC: Certified in Cybersecurity (CC)

ISC2 policies

Admission Policy

Plan to arrive at your test center at least **30 minutes** before your exam start time. To check in for your appointment the following is **required**:

- Show **two (2)** valid, unexpired forms of personal ID (examples include: government issued ids, passports, etc.). Both must have your name (exactly as it appears in your exam registration) and signature, and one of the two must have your photo. For more information about acceptable IDs please visit: https://www.isc2.org/Exams/Exam-Day and look under What You Need to Bring to the Test Center tab for more information.
    - **Unacceptable form of ID: Digital IDs** (A digital ID is an electronic representation of personally identifying information that may be used to verify the identity of a person)

For additional information regarding the Aadhaar ID, please visit Aadhaar ID Policy for Pearson VUE.

- Minors under 18 – Minor must be accompanied by a parent or guardian on the day of exam.
    - Please refer for ID requirements for minors, https://www.isc2.org/exams/exam-day
- Provide your signature.
- Submit to a palm vein scan (unless expressly prohibited by law).
- Have your photo taken. Hats, scarves, and coats may not be worn for your photo. Additionally, you may not wear these items in the test room.
- Leave your personal belongings outside the testing room. You will have access to secure storage. As storage space is limited, please plan appropriately. Pearson VUE test centers do not assume responsibility for your personal belongings.
- Receive a short orientation from the Test Administrator (TA). After the orientation, the TA will escort you to a testing station.
- Sign and agree to the Non-Disclosure Agreement that will be presented at the beginning of your exam. Please take a moment to review the agreement now so that you are familiar with it when you sit for your exam.

Where selected by your Test Sponsor, you agree that Pearson VUE will collect your palm vein pattern at the test center on the day of your exam and retain that information, to the extent permitted by law. Your palm vein scan will be used for the purposes of identification verification on the day of your test and on your future test days, detecting and preventing any fraud, and maintaining the security and integrity of the testing program. For more information on Pearson VUE's policy for use and retention of personal data including biometric data like palm vein scans, please see our Privacy and Cookies Policy. Your agreement to these Testing policies includes agreement to the Privacy and Cookies Policy.

Reschedule Policy

- **If you wish to reschedule** your exam appointment, you must contact Pearson VUE.
- There is no fee for rescheduling the Certified in Cybersecurity (CC) exam. For all other certifications, there is a US$50 fee for exam appointment rescheduled.
- If you choose to **go online to reschedule** your appointment, you must do so **at least 48 hours prior to your appointment**.
- If you choose to **call the Pearson customer support team to reschedule**, you must do so **at least 24 hours prior to your appointment**.

- If you do not reschedule your exam appointment without proper advanced notice, as outlined above, it will result in a no-show, and you will **forfeit your exam fee**. If you used the ISC2 Candidate promo code, as part of the One Million Certified in Cybersecurity initiative, you will **not be able to register again with that code**.
- Once scheduled you have up to 365 days to sit for your exam. Failure to sit for your examination within 365 days will result in a no-show and forfeiture of all exam and rescheduling fees.

Cancellation Policy

- If you wish to cancel your exam appointment, you must contact Pearson VUE.
- There is no fee for canceling the Certified in Cybersecurity (CC) exam. For all other certifications, there is a US$100 fee for exam appointment cancelations.
- If you choose to **go online to cancel** your appointment, you must do so **at least 48 hours prior to your appointment**.
- If you choose to **call the** Pearson customer support team to **cancel**, you must do so **at least 24 hours prior to your appointment**.
- If you do not cancel your exam appointment without proper advanced notice, as outlined above, it will result in a no-show, and will **forfeit your exam fee**. If you used the ISC2 Candidate promo code, as part of the One Million Certified in Cybersecurity initiative, you will **not be able to register again with that code.**

Additional Information

ISC2 Terms and Conditions

- ISC2 requires that all candidates for certification read and accept the terms and conditions set forth here: https://www.isc2.org/uploadedFiles/Certification_Programs/CBT-Examination-Agreement.pdf. Candidates that do not agree to the terms and conditions will not be permitted to sit for any ISC2 examination.

Non-Disclosure Agreement (NDA)

- Failure to read or accept the ISC2 NDA agreement within the allotted five minutes will result in exam termination and forfeiture of exam appointment. Forfeiture of exam appointment also includes forfeiture of all exam fees. To take the examination at a later date you will be required to re-register for the exam and pay all applicable registration fees.

Important Information on ISC2 Exams

- One of the benefits to candidates taking an examination via Computer-Based Testing is that most candidates receive their scores immediately upon completing their examination. In some cases, ISC2 must conduct periodic psychometric analyses prior to releasing exam results. For the small number of candidates affected by this process, it is expected that candidates will receive their results within 6 -8 weeks following the exam.
- ISC2 offers two types of computer-based exams – linear and adaptive – however neither exam type allows for candidates to skip an item, nor can items be returned to later during administration. Once an answer is confirmed it cannot be changed, reviewed, or revisited.
- Frequently asked questions (FAQs) and answers for common inquiries that can be found here: https://www.isc2.org/Frequently-Asked-Questions.

Accommodations Policy

ISC2 provides reasonable and appropriate accommodations for people who have a documented need for exam accommodations. Accommodations must be requested and approved by ISC2 prior to scheduling your examination. If you wish to request an accommodation, please visit https://www.isc2.org/Register-for-Exam and look under the Requesting Special Accommodations tab for information and instructions on how to request an accommodation. Test accommodations are individualized and considered on a case-by-case basis. Once an accommodation is approved, ISC2 will inform the Pearson VUE Accommodations team. Please allow up to three business days for Pearson VUE to receive this information. Then, contact

---

Almost there...

---

back to previous step

Confirm Order Details

Description	Details	Price
Exam CC: Certified in Cybersecurity (CC) Language: English Length: 120 minutes	Appointment Friday, March 14, 2025 Start time: 8:00 AM Africa/Johannesburg - SAST Location Pearson Professional Centers-Johannesburg Pearson VUE 6th Floor Sandton City Office Tower Sandton City Shopping Centre 158 5th Street, SANDTON Johannesburg 2146 South Africa	199.00

Payment Details

Exams for

Name:
Tshingombe Tshingombe Tshitadi
ISC2 ID:
1907033

Order Total

Subtotal:	199.00
Tax:	0.00
TOTAL DUE:	USD 199.00
	USD 199.00

Pearson VUE, so you can schedule your exam, contact information can be found at www.pearsonvue.com/isc2/contact.

Accommodations are not a guarantee of improved performance or exam completion. Once an initial exam appointment is scheduled, there may be a US$50 fee to reschedule an exam with an approved accommodation.

Agree to ISC2 policies

back to previous step

CCSP: Certified Cloud Security Professional (CCSP)

CGRC: Certified in Governance Risk and Compliance

CISSP: Certified Information Systems Security Professional

CSSLP: Certified Secure Software Lifecycle Professional

ISSAP: Information Systems Security Architecture Professional

ISSMP: Information Systems Security Management Professional

SSCP: Systems Security Certified Practitioner

Certification **Exam Outline**

Effective Date: November 15, 2022

About CISSP-ISSMP

The Information Systems Security Management Professional (ISSMP) is a CISSP who specializes in establishing,
presenting and governing information security programs and demonstrates management and leadership
skills. CISSP-ISSMPs direct the alignment of security programs with the organization’s mission, goals and
strategies in order to meet enterprise financial and operational requirements in support of its desired risk
position.
The broad spectrum of topics included in the CISSP-ISSMP Common Body of Knowledge (CBK®) ensure its
relevancy across all disciplines in the field of information security management. Successful candidates are
competent in the following six domains:

- Leadership and Business Management
- Systems Lifecycle Management
- Risk Management
- Threat Intelligence and Incident Management
- Contingency Management

• Law, Ethics and Security Compliance Management

Experience Requirements

Candidates must be a CISSP in good standing and have two years cumulative paid work experience in one or more of the six domains of the CISSP-ISSMP CBK. You can learn more about CISSP-ISSMP experience requirements and how to account for part-time work and internships at www.isc2.org/Certifications/CISSP-Concentrations#steps-to-certification.

Accreditation

CISSP-ISSMP is in compliance with the stringent requirements of ANSI/ISO/IEC Standard 17024.

Job Task Analysis (JTA)

(ISC)² has an obligation to its membership to maintain the relevancy of the CISSP-ISSMP. Conducted at regular intervals, the Job Task Analysis (JTA) is a methodical and critical process of determining the tasks that are performed by security professionals who are engaged in the profession defined by the CISSP-ISSMP. The results of the JTA are used to update the examination. This process ensures that candidates are tested on the topic areas relevant to the roles and responsibilities of today's practicing information security professionals.

CISSP-ISSMP Examination Information

CISSP-ISSMP Examination Weights

Length of exam	3 hours
Number of items	125
Item format	Multiple choice
Passing grade	700 out of 1000 points
Exam availability	English
Testing center	Pearson VUE Testing Center

1. Leadership and Business Management
20%
2. Systems Lifecycle Management
18%

3. Risk Management
19%
4. Threat Intelligence and Incident Management
17%
5. Contingency Management
15%
6. Law, Ethics and Security Compliance Management
11%
Total: 100%

Domain 1:
Leadership and Business Management

1.1 Establish security's role in organizational culture, vision and mission

» Define information security program vision and mission
» Align security with organizational goals, objectives and values
» Define security's relationship to the overall business processes
» Define the relationship between organizational culture and security

1.2 Align security program with organizational governance

» Identify and navigate organizational governance structure
» Validate roles of key stakeholders
» Validate sources and boundaries of authorization
» Advocate and obtain organizational support for security initiatives

1.3 Define and implement information security strategies

» Identify security requirements from business initiatives
» Evaluate capacity and capability to implement security strategies
» Manage implementation of security strategies
» Review and maintain security strategies
» Prescribe security architecture and engineering theories, concepts and methods

1.4 Define and maintain security policy framework Determine applicable external standards

» Determine applicable external standards
» Determine data classification and protection requirements
» Establish internal policies
» Advocate and obtain organizational support for policies
» Develop procedures, standards, guidelines and baselines
» Ensure periodic review of security policy framework

» Define roles and responsibilities
» Determine and manage team accountability
» Build cross-functional relationships
» Resolve conflicts between security and other stakeholders
» Identify communication bottlenecks and barriers
» Integrate security controls into human resources processes
» Evaluate service management agreements (e.g., risk, financial)

» Govern managed services (e.g., infrastructure, cloud services)
» Manage impact of organizational change (e.g., mergers and acquisitions, outsourcing)
» Ensure that appropriate regulatory compliance statements and requirements are included in contractual agreements
» Monitor and enforce compliance with contractual agreements

1.5 Manage security requirements in contracts and agreements

1.6 Manage security awareness and training programs

» Promote security programs to key stakeholders
» Identify needs and implement training programs by target segment
» Monitor and report on effectiveness of security awareness and training programs

1.7 Define, measure and report security metrics

» Identify Key Performance Indicators (KPI)
» Associate Key Performance Indicators (KPI) to the risk posture of the organization
» Use metrics to drive security program development and operations

1.8 Prepare, obtain and administer security budget

» Prepare and secure annual budget
» Adjust budget based on evolving risks and threat landscape
» Manage and report financial responsibilities

1.9 Manage security programs

1.10 Apply product development and project management principles

» Incorporate security into project lifecycle
» Identify and apply appropriate project management methodology
» Analyze project time, scope and cost relationship

2.1 Manage integration of security into Systems Development Life Cycle (SDLC)

» Integrate information security gates (decision points) and requirements into lifecycle
» Implement security controls into system lifecycle
» Oversee security configuration management (CM) processes

2.2 Integrate new business initiatives and emerging technologies into the security architecture

» Integrate security into new business initiatives and emerging technologies
» Address impact of new business initiatives on security posture

2.3 Define and oversee comprehensive vulnerability management programs (e.g., vulnerability scanning, penetration testing, threat analysis)

» Identify, classify and prioritize assets, systems and services based on criticality to business
» Prioritize threats and vulnerabilities
» Manage security testing
» Manage mitigation and/or remediation of vulnerabilities based on risk

2.4 Manage security aspects of change control

» Integrate security requirements with change control process
» Identify and coordinate with the stakeholders
» Manage documentation and tracking
» Ensure policy compliance (e.g., continuous monitoring)

Domain 2:
Systems Lifecycle Management

Domain 3:
Risk Management

3.1 Develop and manage a risk management program
» Identify risk management program objectives
» Communicate and agree on risk management objectives with risk owners and other stakeholders
» Determine scope of organizational risk program
» Identify organizational security risk tolerance/appetite
» Obtain and verify organizational asset inventory
» Analyze organizational risks
» Determine countermeasures, compensating and mitigating controls
» Perform cost-benefit analysis (CBA) of risk treatment options

3.2 Conduct risk assessments
» Identify risk factors

3.3 Manage security risks within the supply chain (e.g., supplier, vendor, third-party risk)
» Identify supply chain security risk requirements
» Integrate supply chain security risks into organizational risk management
» Validate security risk control within the supply chain
» Monitor and review the supply chain security risks

4.1 Establish and maintain threat intelligence program
» Aggregate threat data from multiple threat intelligence sources
» Conduct baseline analysis of network traffic, data and user behavior
» Detect and analyze anomalous behavior patterns for potential concerns
» Conduct threat modeling
» Identify and categorize an attack
» Correlate related security event and threat data
» Create actionable alerting to appropriate resources

4.2 Establish and maintain incident handling and investigation program
» Develop program documentation
» Establish incident response case management process
» Establish incident response team
» Apply incident management methodologies
» Establish and maintain incident handling process
» Establish and maintain investigation process
» Quantify and report financial and operational impact of incidents and investigations to stakeholders
» Conduct root cause analysis (RCA)

Domain 4:
Threat Intelligence and Incident Management

5.1 Facilitate development of contingency plans
» Identify and analyze factors related to the Continuity of Operations Plan (COOP)

» Identify and analyze factors related to the business continuity plan (BCP) (e.g., time, resources, verification)
» Identify and analyze factors related to the disaster recovery plan (DRP) (e.g., time, resources, verification)
» Coordinate contingency management plans with key stakeholders
» Define internal and external crisis communications plans
» Define and communicate contingency roles and responsibilities
» Identify and analyze contingency impact on business processes and priorities
» Manage third-party contingency dependencies
» Prepare security management succession plan

5.2 Develop recovery strategies

» Identify and analyze alternatives
» Recommend and coordinate recovery strategies
» Assign recovery roles and responsibilities

5.3 Maintain contingency plan, Continuity of Operations Plan (COOP), business continuity plan (BCP) and disaster recovery plan (DRP)

» Plan testing, evaluation and modification
» Determine survivability and resiliency capabilities
» Manage plan update process

5.4 Manage disaster response and recovery process

» Declare disaster
» Implement plan
» Restore normal operations
» Gather lessons learned
» Update plan based on lessons learned

Domain 5: Contingency Management

6.1 Identify the impact of laws and regulations that relate to information security

6.2 Adhere to the (ISC)[2] Code of Ethics as related to management issues

6.3 Validate compliance in accordance with applicable laws, regulations and industry best practices

6.4 Coordinate with auditors and regulators in support of the internal and external audit processes

6.5 Document and manage compliance exceptions

» Identify and document compensating controls and workarounds
» Report and obtain authorized approval of risk waiver

Domain 6:

Law, Ethics and Security Compliance Management

» Identify applicable privacy laws
» Identify legal jurisdictions the organization and users operate within (e.g., trans-border data flow)
» Identify export laws
» Identify intellectual property (IP) laws
» Identify applicable industry regulations
» Identify and advise on non-compliance risks
» Inform and advise senior management
» Evaluate and select compliance framework(s)
» Implement the compliance framework(s)
» Define and monitor compliance metrics
» Plan
» Schedule
» Coordinate audit activities
» Evaluate and validate findings
» Formulate response
» Validate implemented mitigation and remediation actions

Additional Examination Information

Supplementary References

Candidates are encouraged to supplement their education and experience by reviewing relevant resources that pertain to the CBK and identifying areas of study that may need additional attention.
View the full list of supplementary references at www.isc2.org/certifications/References.

Examination Policies and Procedures

(ISC)^2 recommends that CISSP-ISSMP candidates review exam policies and procedures prior to registering for the examination. Read the comprehensive breakdown of this important information at www.isc2.org/Exams/Before-Your-Exam.

Legal Info

For any questions related to (ISC)^2's legal policies, please contact the (ISC)^2 Legal Department at legal@isc2.org.

Any Questions?

(ISC)^2 Americas
Tel: +1.866.331.ISC2 (4722)

Email: info@isc2.org
(ISC)₂ Asia-Pacific
Tel: +(852) 28506951
Email: isc2asia@isc2.org
(ISC)₂ EMEA
Tel: +44 (0)203 300 1625
Email: info-emea@isc2.org

v222 Certification **Exam Outline**

Effective Date: November 13, 2020

About CISSP-ISSEP

The Information Systems Security Engineering Professional (ISSEP) is a CISSP who specializes in the practical
application of systems engineering principles and processes to develop secure systems. An ISSEP analyzes
organizational needs, defines security requirements, designs security architectures, develops secure designs,
implements system security, and supports system security assessment and authorization for government and
industry.
The broad spectrum of topics included in the ISSEP Common Body of Knowledge (CBK®) ensure its relevancy
across all disciplines in the field of security engineering. Successful candidates are competent in the following
five domains:

- Systems Security Engineering Foundations
- Risk Management
- Security Planning and Design
- Systems Implementation, Verification and Validation
- Secure Operations, Change Management and Disposal

Experience Requirements

Candidates must be a CISSP in good standing and have two years cumulative paid work experience
in one or more of the five domains of the CISSP-ISSEP CBK. You can learn more about CISSP-ISSEP
experience requirements and how to account for part-time work and internships at www.isc2.org/Certifications/CISSP-ISSEP/experience-requirements.

Accreditation

CISSP-ISSEP is in compliance with the stringent requirements of ANSI/ISO/IEC Standard 17024.

Job Task Analysis (JTA)

(ISC)² has an obligation to its membership to maintain the relevancy of the ISSEP. Conducted at regular
intervals, the Job Task Analysis (JTA) is a methodical and critical process of determining the tasks that are

performed by security professionals who are engaged in the profession defined by the ISSEP. The results of
the JTA are used to update the examination. This process ensures that candidates are tested on the topic
areas relevant to the roles and responsibilities of today’s practicing information security professionals.

CISSP-ISSEP Examination Information

CISSP-ISSEP Examination Weights

Length of exam
3 hours
Number of items
125
Item format
Multiple choice
Passing grade
700 out of 1000 points
Exam availability
English
Testing center
Pearson VUE Testing Center

1. Systems Security Engineering Foundations
25%
2. Risk Management
14%
3. Security Planning and Design
30%
4. Systems Implementation, Verification and Validation
14%
5. Secure Operations, Change Management
and Disposal
17%

Domain 1:
Systems Security Engineering Foundations

1.1 Apply systems security engineering fundamentals
1.2 Execute systems security engineering processes
1.3 Integrate with applicable system development methodology

1.4 Perform technical management
1.5 Participate in the acquisition process
1.6 Design Trusted Systems and Networks (TSN)
» Understand systems security engineering trust concepts and hierarchies
» Identify the relationships between systems and security engineering processes
» Apply structural security design principles
» Integrate security tasks and activities
» Verify security requirements throughout the process
» Integrate software assurance methods
» Perform project planning processes
» Perform project assessment and control processes
» Perform decision management processes
» Perform risk management processes
» Perform configuration management processes
» Perform information management processes
» Perform measurement processes
» Perform Quality Assurance (QA) processes
» Identify opportunities for security process automation
» Identify organizational security authority
» Identify system security policy elements
» Integrate design concepts (e.g., open, proprietary, modular)
» Prepare security requirements for acquisitions
» Participate in selection process
» Participate in Supply Chain Risk Management (SCRM)
» Participate in the development and review of contractual documentation

Domain 2: Risk Management

2.1 Apply security risk management principles
2.2 Address risk to system
2.3 Manage risk to operations
» Establish risk context
» Identify system security risks
» Perform risk analysis
» Perform risk evaluation
» Recommend risk treatment options
» Document risk findings and decisions
» Determine stakeholder risk tolerance
» Identify remediation needs and other system changes
» Determine risk treatment options
» Assess proposed risk treatment options
» Recommend risk treatment options
» Align security risk management with Enterprise Risk Management (ERM)

» Integrate risk management throughout the lifecycle

3.1 Analyze organizational and operational environment
3.2 Apply system security principles
3.3 Develop system requirements
3.4 Create system security architecture and design

Domain 3:
Security Planning and Design

» Capture stakeholder requirements
» Identify relevant constraints and assumptions
» Assess and document threats
» Determine system protection needs
» Develop Security Test Plans (STP)
» Incorporate resiliency methods to address threats
» Apply defense-in-depth concepts
» Identify fail-safe defaults
» Reduce Single Points of Failure (SPOF)
» Incorporate least privilege concept
» Understand economy of mechanism
» Understand Separation of Duties (SoD) concept
» Develop system security context
» Identify functions within the system and security Concept of Operations (CONOPS)
» Document system security requirements baseline
» Analyze system security requirements
» Develop functional analysis and allocation
» Maintain traceability between specified design and system requirements
» Develop system security design components
» Perform trade-off studies
» Assess protection effectiveness

Domain 4:
Systems Implementation, Verification and Validation

4.1 Implement, integrate and deploy security solutions
4.2 Verify and validate security solutions

» Perform system security implementation and integration
» Perform system security deployment activities
» Perform system security verification
» Perform security validation to demonstrate security controls meet stakeholder security requirements

Domain 5:

Secure Operations, Change Management and Disposal

5.1 Develop secure operations strategy
5.2 Participate in secure operations
5.3 Participate in change management
5.4 Participate in the disposal process

» Specify requirements for personnel conducting operations
» Contribute to the continuous communication with stakeholders for security relevant aspects of the system
» Develop continuous monitoring solutions and processes
» Support the Incident Response (IR) process
» Develop secure maintenance strategy
» Participate in change reviews
» Determine change impact
» Perform verification and validation of changes
» Update risk assessment documentation
» Identify disposal security requirements
» Develop secure disposal strategy
» Develop decommissioning and disposal procedures
» Audit results of the decommissioning and disposal process

Additional Examination Information

Supplementary References

Candidates are encouraged to supplement their education and experience by reviewing
relevant resources that pertain to the CBK and identifying areas of study that may need
additional attention.
View the full list of supplementary references at
www.isc2.org/certifications/References.

Examination Policies and Procedures

(ISC)² recommends that ISSEP candidates review exam policies and procedures prior to
registering for the examination. Read the comprehensive breakdown of this important
information at www.isc2.org/Register-for-Exam.

Legal Info

For any questions related to (ISC)²'s legal policies, please contact the (ISC)₂ Legal Department at legal@isc2.org.

Any Questions?

(ISC)² Americas
Tel: +1-866-331-ISC2(4722)

Email: membersupport@isc2.org
(ISC)² Asia Pacific
Tel: +852-2850-6951
Email: membersupportapac@isc2.org
(ISC)² EMEA
Tel: +44-203-960-7800
Email: membersupportemea@isc2.org

Attachments

Rate this

Details

Lenovo and Intel are Driving AI Innovation at the Edge

Flynn Maloy, Chief Marketing Officer of Lenovo ISG

Jan 23 2025| 0 mins

Lenovo and Intel's long-standing partnership is transforming industries by bringing cutting-edge AI solutions to the edge and beyond. From PCs to data centers, our collaboration has consistently pushed technological boundaries. The strength of Lenovo's ThinkEdge portfolio is enabling AI-driven applications in manufacturing sites, retail stores, schools, and more. Join @Flynn Maloy, Chief Marketing Officer of Lenovo ISG, as he details how Lenovo and Intel® are leading the way in AI innovation: - Comprehensive solutions for diverse industries: From computer vision in manufacturing to advanced AI in education and retail, Lenovo and Intel's joint solutions empower a variety of applications. - Next-gen AI with CPUs: Not every AI workload requires massive GPUs. Intel's CPUs are driving the next wave of edge AI, particularly in inferencing and delivering efficient and accessible AI solutions. - Scalable and powerful edge portfolio: Lenovo's edge clients and servers, powered by Intel, are designed to meet the demands of modern businesses, offering flexibility and performance across workloads. - A partnership that drives innovation: With a shared vision for the future of AI, Lenovo and Intel continue to push the boundaries of what's possible for our customers. Together, Lenovo and Intel are leading the charge in making AI more accessible, scalable, and impactful for businesses worldwide.

State of Cloud 2025: Navigating EMEA's Cloud Revolution

John Bradshaw, Director of Cloud Computing Technology and Strategy, EMEA, Akamai & Bryan Glick, Editor in Chief, Computer Weekly

Feb 27 2025| 18 mins

Boris Cipot, Senior Security Engineer

Sep 05 2024| 30 mins

Python is a fast, platform-agnostic, and easy-to-learn programming language that is suited for beginners and experienced developers alike. Ever since its first release in 1991, Python has had a constant presence in the computer world and has become a go-to language thanks to its easy-to-understand code and versatility. Today, Python can boast a wide array of libraries and frameworks, and they are the cornerstone of fast and easy Python programming—the so-called Pythonic way of development. But like all programming languages, Python is not immune to security threats. Secure coding best practices must be adopted to avoid risks from attackers. In this webinar, we'll explore Python security best practices that should employed when building secure application.

One-Stop DevOps: Simplifying Toolchains with GitLab and Google Cloud

Nate Avery, Outbound Product Manager - Google | Jackie Porter, Director of Product - Gitlab | Torsten Volk, Principal Analyst - ESG

Dec 04 2024| 28 mins

Seamless Edge Deployment and Management with Lenovo and Intel

Blake Kerrigan, Senior Director, ThinkEdge Business Group

Jan 23 2025| 1 mins

Sort by

Career Opportunity

Senior Applied Scientist – Copilot Team

Posted: March 3, 2025

Location: Beijing, China

Research Area(s): Artificial intelligence

We are inviting you to join the Copilot Team, where we are redefining the future of AI-powered experiences. The Copilot Team is at the forefront of innovation, building intelligent solutions that empower users across devices…

Career Opportunity

Senior Applied AI Engineer – Microsoft Security AI Research team

Posted: March 3, 2025

Location: Remote (within US)

Research Area(s): Artificial intelligence, Security, privacy, and cryptography

Join the vanguard of cybersecurity innovation with the Microsoft Security AI Research team. We are on the lookout for an Applied Scientist to spearhead the research and development of functional autonomous agents for security scenarios.…

Career Opportunity

Data Scientist II – Microsoft Security

Posted: March 1, 2025

Location: Remote (within US); United States

Research Area(s): Artificial intelligence, Data platforms and analytics, Human-computer interaction, Security, privacy, and cryptography

The AI Personalization, Feedback, and Analytics team ensures that Security Copilot, Microsoft's GenAI platform, delivers adaptive and intelligent experiences by leveraging feedback loops, analytics, and personalization techniques. We are seeking a Data Scientist to help…

Career Opportunity

Senior Applied Scientist – Power Apps

Posted: March 1, 2025

Location: Redmond, WA, US; Remote (within US)

Research Area(s): Algorithms, Artificial intelligence, Data platforms and analytics

The Power Apps team at Microsoft is looking to hire a Senior Applied Scientist. As a team, we are very customer focused and driven by curiosity, creativity, teamwork, agility, accountability and desire to learn everyday.…

Career Opportunity

Applied Scientist II – Power Apps

Posted: March 1, 2025

Location: Redmond, WA, US; Remote (within US)

Research Area(s): Algorithms, Artificial intelligence, Data platforms and analytics, Programming languages and software engineering

The Power Apps team at Microsoft is looking to hire an Applied Scientist II. As a team, we are very customer focused and driven by curiosity, creativity, team work, agility, accountability and desire to learn everyday. If…

Career Opportunity

Principal Applied Scientist – Advanced Autonomy and Applied Robotics

Posted: March 1, 2025

Location: Redmond, WA, US

Research Area(s): Artificial intelligence, Hardware and devices, Human-computer interaction, Technology for emerging markets

Within Microsoft’s Strategic Missions and Technologies (SMT) division, the Advanced Autonomy and Applied Robotics team is seeking a Principal Applied Scientist.The role involves building the future platform for human-robot-agent teaming. This individual will leverage cutting-edge AI and robotics technologies…

Career Opportunity

Senior Applied Scientist – Advanced Autonomy and Applied Robotics

Posted: March 1, 2025

Location: Redmond, WA, US

Research Area(s): Artificial intelligence, Hardware and devices, Human-computer interaction, Technology for emerging markets

Within Microsoft’s Strategic Missions and Technologies (SMT) division, the Advanced Autonomy and Applied Robotics team is seeking a Senior Applied Scientist. The role involves building the future platform for human-robot-agent teaming. This individual will leverage…

Career Opportunity

Principal Researcher – Generative AI – Microsoft Research AI Frontiers

Posted: March 1, 2025

Location: New York, NY, US; Redmond, WA, US

Research Area(s): Artificial intelligence

We are seeking a Principal Researcher to join our team and lead efforts on the advancement of Generative AI and Large Language Models (LLMs) technologies. As a Principal Researcher, you will play a crucial role in leading,…

Career Opportunity

Senior Applied Scientist

Posted: March 1, 2025

Location: Cairo, Egypt

Research Area(s): Artificial intelligence

In shaping the future of monetization for personalized AI assistants and pioneering innovation in the advertiser agentic space, as a Senior Applied Scientist, you will collaborate with engineers, data scientists, and product managers to develop…

Career Opportunity

Principal Data Scientist – Real-Time Intelligence team

Posted: February 28, 2025

Location: Redmond, WA, US

Research Area(s): Artificial intelligence, Data platforms and analytics, Systems and networking

Microsoft Fabric’s Real-Time Intelligence team is leading the transformation of real-time analytics in the world of data. We are hiring a Principal Data Scientist to tackle challenges in both open-source and proprietary technologies related to

---

engineering

Inbox

tshingombe fiston <tshingombefiston@gmail.com>

Mon, Mar 3, 3:19 PM (18 hours ago)

to
me

namics 365 Community / My Profile

CU03031227-0

Stats

0 Comments
0 Posts
1 Likes
0 Questions
My activity

Achievements

-

Personal information

- Achievements
- Notifications
- Notification settings
- Quick responses

Personal information

Email

tshingombefiston@gmail.com

Confirmation

Registration details

Name

tshingombe tshitadi

Status

Registered

Registration ID

102231646

Quick Links

- Go to the Microsoft Research Forum Website
- Registration support
- Cancel registration

Please note: event emails will be sent to the email address you provided during registration. If you are not receiving event communications, please check your 'junk', 'spam', or 'clutter' folders to confirm your email settings have not redirected the emails. In addition, please add msresearchforum@eventcore.com to your 'safe sender' list to ensure you receive future communications for this event.

Microsoft is committed to your privacy. If you have questions surrounding how your registration is affected by the General Data Protection Regulation (GDPR), then please visit Privacy Management for more information.

Share the news you've registered!

Compare Microsoft 365 with Office

Best value for 2 to 6 people

Microsoft 365 Family

Subscription
R1 999,00/year
Buy now
Or buy at R199,00/month
Try free for 1 month

Microsoft 365 Personal

Subscription
R1 599,00/year

Buy now
Or buy at R159,00/month

Office Home 2024

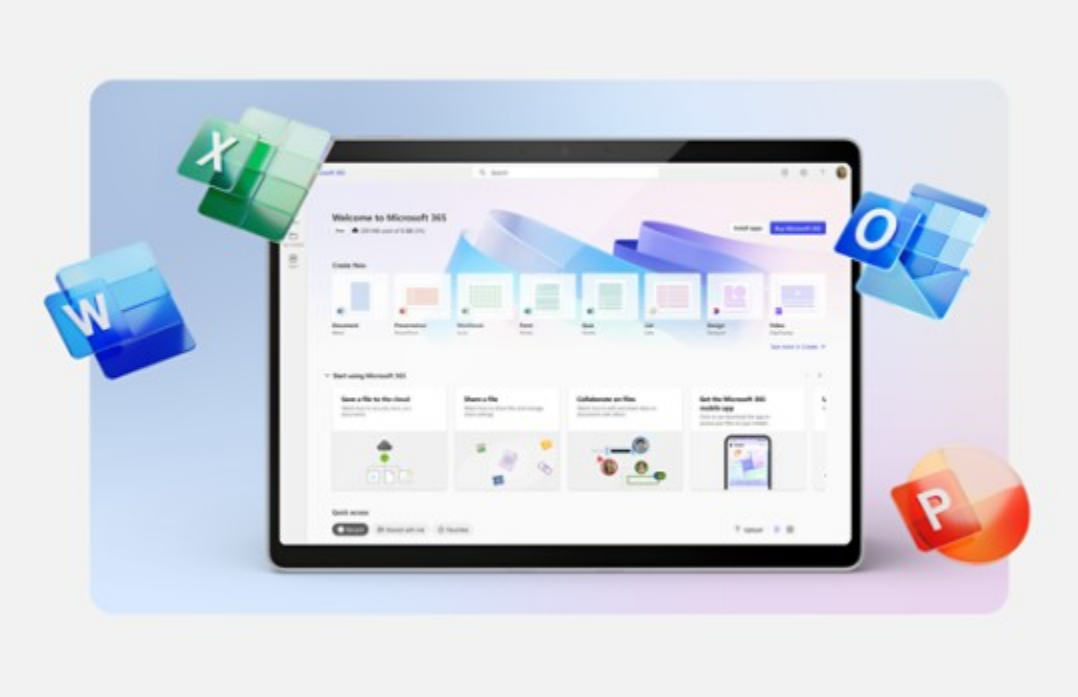
One-time purchase for PC or Mac
R2 299,00
Buy now

	Microsoft 365 Family	Microsoft 365 Personal	Office Home 2024
Number of users	1 to 6 people	1 person	One PC or Mac
Number of accounts	1 to 6	1	One PC or Mac
Cloud storage	Up to 6 TB (1 TB per person)	1 TB	
Multiple devices and platforms	Included	Included	
Word, Excel, PowerPoint	Included	Included	Included
OneNote	Included	Included	Included
Outlook	Included	Included	
Ongoing technical support	Included	Included	
Microsoft Defender	Included	Included	
Microsoft Editor	Included	Included	
Clipchamp	Included	Included	
Microsoft Teams	Included	Included	
Access (PC only)	Included	Included	
Microsoft Forms	Included	Included	
	• Learn more	• Learn more	• Learn more

Office Home & Business 2024

- One-time purchase for one PC or Mac
- Classic 2024 desktop versions of Word, Excel, PowerPoint, Outlook and OneNote
- Access to support resources

Buy now Learn more



Frequently asked questions

| 4 is sold as a one-time purchase, which means you pay a single, up-front cost to get Office apps for one computer. One-time purchases are available for both PCs and Macs. However, there are no upgrade options, which means if you plan to upgrade to the next major release, you'll have to buy it at full price.

Microsoft 365 Personal and Microsoft 365 Family are subscriptions that include powerful productivity apps and creativity tools with AI-powered features. In addition to premium desktop versions of popular Microsoft 365 apps like Word, PowerPoint, Excel and Outlook, you also get spacious cloud storage and cloud-connected features that let you collaborate on files in real time. With a subscription, you'll always have the latest features, fixes and security updates along with ongoing tech support at no extra cost. You can choose to pay for your subscription on a monthly or yearly basis, and use your apps on multiple PCs, Macs, tablets and phones. Additionally, the Microsoft 365 Family plan lets you share your subscription with up to five more people. Everyone gets their own apps and storage. (AI features only available to subscription owner and cannot be shared; AI usage limits apply; minimum age limits may apply to subscription activation and use of AI features. Learn more.)

Microsoft 365 is compatible with PC, Mac, Android and iOS.
See system requirements for compatible versions of your devices, and for other feature requirements.

 No. Microsoft 365's applications are tailored for each platform and each operating system. The applications available for Mac users and the specific features included may be different from those available for PC users. With Microsoft 365, you can be flexible. With your account, you are not limited to exclusively Mac or exclusively PC, so you can transition across devices.
 Yes. Documents that you have created belong fully to you. You can choose to store them online on OneDrive or locally on your PC or Mac.

 Internet access is required to install and activate all the latest releases of apps and services included in all Microsoft 365 subscription plans. Note that if you are an existing subscriber, you do not need to reinstall or purchase another subscription.

For Microsoft 365 plans, internet access is also needed to manage your subscription account, for example to install Office apps on other PCs or to change billing options. Internet access is also required to access documents stored on OneDrive, unless you install the OneDrive desktop app.

You should also connect to the internet regularly to keep your version of Microsoft 365 up to date and to benefit from automatic upgrades. If you do not connect to the internet at least every 31 days, your apps will go into reduced functionality mode, which means that you can view or print your documents but cannot edit the documents or create new ones. To reactivate your apps, simply reconnect to the internet.

You do not need to be connected to the internet to use the Office apps, such as Word, Excel and PowerPoint, because the apps are fully installed on your computer.

 Your Microsoft account is the combination of an email address and password that you use to sign in to services like OneDrive, Xbox LIVE and Outlook.com. If you use any of these services, you already have a Microsoft account that you can use, or you can create a new account. Learn more about Microsoft accounts.

As part of signing up for a trial or purchasing Microsoft 365, you will be prompted to sign in with a Microsoft account. You must be signed in with this account to install and manage your Microsoft 365 subscription, or to use some subscription benefits, including cloud storage.

 You can share Microsoft 365 Family with five other people, for a total of six users. Microsoft 365 Personal can be used by one person.

 If you have an active Microsoft 365 Family subscription, you can share it with up to five other people. Each person you share your subscription

with can install Microsoft 365 on all their devices and sign in to five devices at the same time.

To add someone to your subscription, sign in to your Microsoft account and follow the on-screen instructions to add a user. Each person you add will receive an email with the steps they need to follow. Once they have accepted and completed the steps, their information, including the installs they are using, will appear on their My Account page. You can stop sharing your subscription with someone or remove a device they are using by logging into your Microsoft account.

 Visit learn more about free apps.
 Microsoft Defender is a cross-device security app that helps individuals and families protect their data and devices by continuously scanning the web for threats to your identity and personal data (US only). Defender also helps you stay safer online with malware protection, real-time security notifications and security tips. Download the Microsoft Defender app.

 Microsoft Defender is a new cross-device app that helps people and families stay safer online. Microsoft Defender adds new features and a simplified user interface. Microsoft Defender also brings valuable device protection to iOS, Android, Windows and Mac, with malware protection, web protection, real-time security notifications and security tips. Microsoft Defender is available in the Apple, Google and Microsoft app stores and requires a Microsoft 365 Personal or Family subscription to use.

Windows Security, formerly known as Windows Defender Security Centre, is built-in security on Windows PCs to protect your device and data. Windows Security is pre-installed and automatically enabled. Windows Security includes Microsoft Defender Antivirus software that protects your Windows device and data against viruses, ransomware, trojans and other malware unless non-Microsoft antivirus software is active.

 A free in-browser video editing platform designed to make video creation accessible for everyone.
 AI features included in Microsoft 365 Family plans are only available to the subscription owner and cannot be shared with others.
 To use Copilot in Word, Excel, PowerPoint, OneNote and Outlook, make sure you have the latest version of Microsoft 365 installed. If you're signed in, have the latest updates installed, and still don't see Copilot, please restart your Microsoft 365 apps. Learn more about why I am not seeing Copilot in my apps.
 Microsoft 365 supports Arabic, Chinese Simplified, Chinese Traditional, Czech, Danish, Dutch, English, Finnish, French, German, Hebrew, Hungarian, Italian, Japanese, Korean, Norwegian, Polish, Portuguese, Russian, Spanish, Swedish, Thai and Turkish. Some Designer

features, like inline editing capabilities, are available only in English. We plan to add more languages soon. You can also learn more about Copilot supported languages here: Copilot for Microsoft 365 supported languages – Microsoft Support.
 Visit our Copilot help & learning site to start using Copilot today.
 Microsoft Designer is a graphic design and image editing app powered by AI. Create eye-catching images with your words, craft next-level designs that pop and even edit photos like an expert. Designer is integrated across your favourite Microsoft apps like Word and PowerPoint to help you create when and where you need it.
 Beyond the Microsoft Designer web and mobile app, certain Designer features are integrated across some of your favourite Microsoft apps like Word and PowerPoint, helping spark creativity where and when you need it. For Windows users, Designer is also integrated into Microsoft Photos.
 Usage limits apply to AI-powered features, including Copilot and Designer. Your Microsoft 365 Personal or Family subscription unlocks AI credits to experience and engage with Copilot across Microsoft 365 apps and beyond. Learn more about credits.
 Microsoft 365 Business Basic, Business Standard and Business Premium are tailored for businesses, offering professional email with a custom domain, admin controls for managing access and devices and scalability to add additional users as your business grows. They include advanced security features like Exchange Online Protection to guard against phishing and malware, with Business Premium adding Microsoft Defender for Business for ransomware protection and advanced threat management. Plus, you can access professional collaboration tools like Microsoft Teams with meeting recordings, transcription and team workspaces, while business apps such as Microsoft Bookings can simplify meeting and appointment scheduling. Additionally, Microsoft 365 Copilot, an AI-powered assistant for work, is available as an add-on to boost productivity and creativit

---

Course

Microsoft Azure AI Fundamentals

Course AI-900T00-A: Microsoft Azure AI Fundamentals

At a glance

- Level

  Beginner

- Product

  Azure

- Role

  AI Engineer

- Languages

  English Arabic Chinese (Simplified) Chinese (Traditional) French German Indonesian Italian Japanese Korean Portuguese (Brazil) Russian Spanish

- Course Duration

  1 day

- Related certifications

  Microsoft Certified: Azure AI Fundamentals

Overview

This course introduces fundamentals concepts related to artificial intelligence (AI), and the services in Microsoft Azure that can be used to create AI solutions. The course is not designed to teach students to become professional data scientists or software developers, but rather to build awareness of common AI workloads and the ability to identify Azure services to support them. The course is designed as a blended learning experience that combines instructor-led training with online materials on the Microsoft Learn platform (https://azure.com/learn). The hands-on exercises in the course are based on Learn modules, and students are encouraged to use the content on Learn as reference materials to reinforce what they learn in the class and to explore topics in more depth.

Audience Profile

The Azure AI Fundamentals course is designed for anyone interested in learning about the types of solution artificial intelligence (AI) makes possible, and the services on Microsoft Azure that you can use to create them. You don’t need to have any experience of using Microsoft Azure before taking this course, but a basic level of familiarity with computer technology and the Internet is assumed. Some of the concepts covered in the course require a basic understanding of mathematics, such as the ability to interpret charts. The course includes hands-on activities that involve

working with data and running code, so a knowledge of fundamental programming principles will be helpful.

Course Syllabus

You can prepare in instructor-led training or self-paced study

- Learning Path

  Microsoft Azure AI Fundamentals: AI Overview

  - 3 Modules
  - Beginner
  - AI Engineer
  - Azure AI Bot Service

70%

-

 Learning Path

Microsoft Azure AI Fundamentals: Computer Vision

- 3 Modules

- Beginner
- AI Engineer
- Azure

97%

-

 Learning Path

Microsoft Azure AI Fundamentals: Natural Language Processing

- 5 Modules

- Beginner
- AI Engineer
- Azure Portal

95%

-

 Learning Path

Microsoft Azure AI Fundamentals: Document Intelligence and Knowledge Mining

- 2 Modules

- Beginner
- AI Engineer
- Azure

Completed

-

 Learning Path

Microsoft Azure AI Fundamentals: Generative AI

- 1 of 4 modules completed

- Beginner
- AI Engineer
- Azure OpenAI Service

20%

-

-

Search for a training provider

Course

Microsoft Azure AI Fundamentals

Course AI-900T00-A: Microsoft Azure AI Fundamentals

At a glance

- Level

  Beginner

- Product

  Azure

- Role

  AI Engineer

- Languages

  English Arabic Chinese (Simplified) Chinese (Traditional) French German Indonesian Italian Japanese Korean Portuguese (Brazil) Russian Spanish

- Course Duration

  1 day

- Related certifications

  Microsoft Certified: Azure AI Fundamentals

Overview

This course introduces fundamentals concepts related to artificial intelligence (AI), and the services in Microsoft Azure that can be used to create AI solutions. The course is not designed to teach students to become professional data scientists or software developers, but rather to build awareness of common AI workloads and the ability to identify Azure services to support them. The course is designed as a blended learning experience that combines instructor-led training with online materials on the Microsoft Learn platform (https://azure.com/learn). The hands-on exercises in the course are based on Learn modules, and students are encouraged to use the content on Learn as reference materials to reinforce what they learn in the class and to explore topics in more depth.

Audience Profile

The Azure AI Fundamentals course is designed for anyone interested in learning about the types of solution artificial intelligence (AI) makes possible, and the services on Microsoft Azure that you can use to create them. You don’t need to have any experience of using Microsoft Azure before taking this course, but a basic level of familiarity with computer technology and the Internet is assumed. Some of the concepts covered in the course require a basic understanding of mathematics, such as the ability to interpret charts. The course includes hands-on activities that involve working with data and running code, so a knowledge of fundamental programming principles will be helpful.

Course Syllabus

You can prepare in instructor-led training or self-paced study

- Learning Path

  Microsoft Azure AI Fundamentals: AI Overview

  - 3 Modules
  - Beginner
  - AI Engineer
  - Azure AI Bot Service

70%

-

 Learning Path

Microsoft Azure AI Fundamentals: Computer Vision

- 3 Modules

- Beginner
- AI Engineer
- Azure

97%

-

Learning Path

Microsoft Azure AI Fundamentals: Natural Language Processing

- 5 Modules

- Beginner
- AI Engineer
- Azure Portal

95%

-

Learning Path

Microsoft Azure AI Fundamentals: Document Intelligence and Knowledge Mining

- 2 Modules

- Beginner
- AI Engineer
- Azure

Completed

-

Learning Path

Microsoft Azure AI Fundamentals: Generative AI

- 1 of 4 modules completed

- Beginner
- AI Engineer
- Azure OpenAI Service

20%

Course

Microsoft Azure AI Fundamentals

Course AI-900T00-A: Microsoft Azure AI Fundamentals

At a glance

- Level

  Beginner

- Product

  Azure

- Role

  AI Engineer

- Languages

  English Arabic Chinese (Simplified) Chinese (Traditional) French German Indonesian Italian Japanese Korean Portuguese (Brazil) Russian Spanish

- Course Duration

  1 day

- Related certifications

  Microsoft Certified: Azure AI Fundamentals

Overview

This course introduces fundamentals concepts related to artificial intelligence (AI), and the services in Microsoft Azure that can be used to create AI solutions. The course is not designed to teach students to become professional data scientists or software developers, but rather to build awareness of common AI workloads and the ability to identify Azure services to support them. The course is designed as a blended learning experience that combines instructor-led training with online materials on the Microsoft Learn platform (https://azure.com/learn). The hands-on exercises in the course are based on Learn modules, and students are encouraged to use the content on Learn as reference materials to reinforce what they learn in the class and to explore topics in more depth.

Audience Profile

The Azure AI Fundamentals course is designed for anyone interested in learning about the types of solution artificial intelligence (AI) makes possible, and the services on Microsoft Azure that you can use to create them. You don’t need to have any experience of using Microsoft Azure before taking this course, but a basic level of familiarity with computer technology and the Internet is assumed. Some of the concepts covered in the course require a basic understanding of mathematics, such as the ability to interpret charts. The course includes hands-on activities that involve working with data and running code, so a knowledge of fundamental programming principles will be helpful.

Course Syllabus

You can prepare in instructor-led training or self-paced study

- Learning Path

  Microsoft Azure AI Fundamentals: AI Overview

  - 3 Modules
  - Beginner
  - AI Engineer
  - Azure AI Bot Service

70%

-

 Learning Path

Microsoft Azure AI Fundamentals: Computer Vision

- 3 Modules

- Beginner
- AI Engineer
- Azure

97%

-

Learning Path

Microsoft Azure AI Fundamentals: Natural Language Processing

- 5 Modules

- Beginner
- AI Engineer
- Azure Portal

95%

-

Learning Path

Microsoft Azure AI Fundamentals: Document Intelligence and Knowledge Mining

- 2 Modules

- Beginner
- AI Engineer
- Azure

Completed

-

Learning Path

Microsoft Azure AI Fundamentals: Generative AI

- 1 of 4 modules completed

- Beginner
- AI Engineer
- Azure OpenAI Service

20%

1000 XP

Embrace responsible AI principles and practices

- 51 min
- Module
- 9 Units

Beginner
Business Owner
Business User
Azure
Dynamics 365
Microsoft 365

This module is designed to help you adopt responsible AI practices. It offers an overview of the principles, governance system, and procedures followed at Microsoft, but we encourage you to develop your own AI strategy.

Learning objectives

In this module, you will:

- Prepare for the implications of responsible AI
- Describe principles of responsible AI
- Establish a system for AI governance
- Take actions for AI governance
- Engage across teams and organizations to implement responsible AI principles
- Take inspiration from ho

Fundamentals of Generative AI

- 9 min remaining
- Module
- 11 Units

Beginner
AI Engineer
Developer
Solution Architect
Student
Azure OpenAI Service

Azure

In this module, you explore the way in which language models enable AI applications and services to generate original content based on natural language input. You also learn how generative AI enables the creation of agents that can assist humans in creative tasks.

Learning objectives

By the end of this module, you are able to:

- Understand generative AI's place in the development of artificial intelligence.
- Understand language models and their role in intelligent applications.
- Describe examples of Microsoft Copilot, agents, and good prompts.

Knowledge check

Completed 200 XP

- Module assessment
- 3 minutes

1.

What are Large Language Models?

Models that detect additional meaning in paragraphs of text.

Lists of words and code that computers use to generate text.

Models that use deep learning to process and understand natural language on a massive scale.

2.

Which Microsoft Copilot should a customer support agent use to research and resolve a support issue?

Microsoft Copilot for Microsoft Edge

Microsoft Copilot in Dynamics 365 Customer Service

Copilot for Security

3.

Which tool should a professional developer use to build a custom copilot and deploy it as a service endpoint in Azure?

Copilot for Azure

Microsoft Copilot Studio

Microsoft Azure AI Foundry

All units complete:

Having an issue? We can help!

3.

3900 XP

Craft effective prompts for Microsoft 365 Copilot

- 2 hr 10 min
- Learning Path
- 0 of 4 modules completed

At a glance

- Level

  Beginner

- Skill

  Create effective prompts for Microsoft Copilot for Microsoft 365

- Product

  Microsoft Copilot Microsoft 365 Microsoft 365 Apps Word PowerPoint Excel Outlook Microsoft Teams OneNote

- Role

  Business User

- Subject

  Business applications Productivity Artificial intelligence

Discover ways to craft effective and contextual prompts for Microsoft 365 Copilot that create, simplify, transform, and compile content across Microsoft 365 applications. Learn the importance of providing a clear goal, context, source, and expectation in your prompt for the best results. This course covers real world scenarios and examples using Copilot in Microsoft 365 apps like Word, Excel, PowerPoint, Teams, Outlook, OneNote, and Chat.

Note

This content was partially created with the help of AI. An author reviewed and revised the content as needed. Read more.

Prerequisites

Learners should have completed the following content prior to this course:

- Fundamentals of Generative AI
- Get started with Microsoft 365 Copilot

Developer

Accelerate app development by using GitHub Copilot

Find out how to use GitHub Copilot to interpret and document code, author new code features more efficiently, and refactor, debug, and test code.

Build AI apps with Azure Services and best practices

Get the details on designing and building a cloud-native AI app, developing a back-end database, and integrating Azure AI services into applications.

Build and extend copilots with Microsoft Copilot Studio

Use Microsoft Copilot Studio to create conversational AI solutions, and learn how to build actions that extend Microsoft 365 Copilot.

Extend Microsoft 365 Copilot (for developers)

Use Copilot Studio actions, and learn about building plugins and connectors for Microsoft 365 Copilot. Discover how to choose the right option for your u

Business or technical leader

Transform your business with Microsoft AI

In this learning path, business leaders will find the knowledge and resources to adopt AI in their organizations. Explore planning, strategizing, and scaling AI projects in a responsible way.

Implement data integration and model grounding with Azure AI Foundry and Microsoft Fabric

Discover how to create advanced AI solutions, ground models in their data, connect and integrate data from various sources, and use OneLake in Microsoft Fabric.

Accelerate gen AI model selection, evaluation, and multimodal integration with Azure AI Foundry

Find out how to benchmark models, apply multimodal models to help enhance customer satisfaction, and complete evaluations to help ensure performance and safety.

Unlocking business potential with AI solutions

Learn how to initiate your organization's AI strategy, assess infrastructure readiness, and understand the business impact of AI

Business user

Design a dream destination using Microsoft Copilot

Bring your personal creativity and passion to dream up a novel destination and create the content to help tell its story. Interact with Microsoft Copilot to learn about the capabilities of generative AI.

Build your Microsoft 365 Copilot skills (for end users)

Find out how to create effective prompts in Microsoft 365 Copilot to help boost your productivity. Explore real-world prompts for specific use case scenarios.

Work smarter with AI

Get more done and unleash your creativity with Microsoft Copilot. In this learning path, you'll explore how to use Microsoft Copilot to help you research, find information, and generate

Data scientist

Make your data AI ready with Microsoft Fabric

Discover how to implement large-scale data engineering, lakehouse, and warehouse solutions using Microsoft Fabric. Build the skills to use Fabric to effectively manage and analyze data.

Run data analytics solutions with Azure Databricks

Work with Apache Spark and Azure Databricks to run large data engineering workloads in the cloud, and use Azure Databricks for comprehensive data analytics solutions.

IT professional

Get AI-Ready with Microsoft 365 Admin

This content helps admins ensure that Microsoft 365 tenants are set up and configured for AI so that future AI features can be integrated as seamlessly as possible.

Discover Microsoft 365 Copilot (for administrators)

Focus on security and compliance features to configure in your Microsoft 365 tenant to help protect your organizational data before you implement Microsoft 365 Copilot.

Low-code developer

Create Power Platform solutions with AI and Copilot

Learn to use Copilot to set up Dataverse, create Power Apps, and build Automated Processes. Explore what Microsoft Copilot Studio can do to help you build and extend custom copilots.

Accelerate AI development with Low Code

Learn how to develop on Dataverse, Power Apps, and Power Automate. This curated content will also cover creation of custom copilots with Microsoft Copilot Studio.

Extend Microsoft 365 Copilot (for developers)

Use Copilot Studio actions, and learn about building plugins and connectors for Microsoft 365 Copilot. Discover how to choose the right option for your use case.

Build and extend copilots with Microsoft Copilot Studio

Use Microsoft Copilot Studio to create conversational AI solutions, and learn how to build actions that extend Microsoft 365 Copilot.

ecurity professional

Help secure your data in the age of AI

Work with Microsoft Purview, Microsoft Sentinel, and Microsoft Copilot for Security, and learn how to effectively manage, protect, and govern sensitive information in AI-driven environments.

Plan

Help secure your data in the age of AI

3 milestones

This Plan is designed to offer you interactive experience working with Microsoft technologies, including Microsoft Purview, Microsoft Sentinel, and Microsoft Copilot for Security, so you can effectively manage, protect, and govern sensitive information in AI-driven environments. Discover how to create a secure and compliant data estate that can easily adapt to AI.• Access Control and Identity Management, 3rd Ed. by Mike Chapple. Publisher: Jones and Bartlett Learning. (Sep, 2020). _____ • Building an Informati

- Published on 3/4/2025
- Created by 46307064

Accelerate app development by using GitHub Copilot

3 milestones

This Plan is designed to help you enhance your coding efficiency and accuracy. Find out how to use GitHub Copilot to interpret and document code, so you can quickly ramp up on unfamiliar or complex codebases. Learn to author new code features more efficiently by using GitHub Copilot autocompletion and chat features. Additionally, get the details on refactoring, debugging, and testing code with GitHub Copilot.

- Published on 3/4/2025
- Created by 46307064
- **Tell us about your PDF experience.**

Install C and C++ support in Visual Studio

Article • 12/09/2021

If you haven't downloaded and installed Visual Studio and the Microsoft C/C++ tools yet, here's how to get started.

Visual Studio 2022 Installation

Welcome to Visual Studio 2022! In this version, it's easy to choose and install just the features you need. And because of its reduced minimum footprint, it installs quickly and with less system impact.

Note

This topic applies to installation of Visual Studio on Windows. **Visual Studio Code** is a lightweight, cross-platform development environment that runs on Windows, Mac, and Linux systems. The Microsoft **C/C++ for Visual Studio Code** extension supports IntelliSense, debugging, code formatting, auto-completion. Visual Studio for Mac doesn't support Microsoft C++, but does support .NET languages and cross-platform development. For installation instructions, see **Install Visual Studio for Mac**.

Want to know more about what else is new in this version? See the Visual Studio release notes.

Ready to install? We'll walk you through it, step-by-step.

Step 1 - Make sure your computer is ready for Visual Studio

Before you begin installing Visual Studio:

1. Check the system requirements. These requirements help you know whether your computer supports Visual Studio 2022.
2. Apply the latest Windows updates. These updates ensure that your computer has

- both the latest security updates and the required system components for Visual
- Studio.3. Reboot. The reboot ensures that any pending installs or updates don't hinder the
- Visual Studio install.
- 4. Free up space. Remove unneeded files and applications from your %SystemDrive%
- by, for example, running the Disk Cleanup app.
- For questions about running previous versions of Visual Studio side by side with Visual
- Studio 2022, see the Visual Studio 2022 Platform Targeting and Compatibility page.

Step 2 - Download Visual Studio

- Next, download the Visual Studio bootstrapper file. To do so, choose the following
- button to go to the Visual Studio download page. Select the edition of Visual Studio that
- you want and choose the **Free trial** or **Free download** button.

Step 3 - Install the Visual Studio installer

- Run the bootstrapper file you downloaded to install the Visual Studio Installer. This new
- lightweight installer includes everything you need to both install and customize Visual
- Studio.
- 1. From your **Downloads** folder, double-click the bootstrapper that matches or is
- similar to one of the following files:
- **vs_community.exe** for Visual Studio Community
- **vs_professional.exe** for Visual Studio Professional
- **vs_enterprise.exe** for Visual Studio Enterprise
- If you receive a User Account Control notice, choose **Yes** to allow the bootstrapper
- to run.
- 2. We'll ask you to acknowledge the Microsoft License Terms and the Microsoft
- Privacy Statement . Choose **Continue**.

Step 4 - Choose workloads

- After the installer is installed, you can use it to customize your installation by selecting
- the *workloads*, or feature sets, that you want. Here's how.
- 1. Find the workload you want in the **Installing Visual Studio** screen.For core C and C++ support, choose the "Desktop development with C++"

- workload. It comes with the default core editor, which includes basic code editing
- support for over 20 languages, the ability to open and edit code from any folder
- without requiring a project, and integrated source code control.
- Additional workloads support other kinds of development. For example, choose
- the "Universal Windows Platform development" workload to create apps that use
- the Windows Runtime for the Microsoft Store. Choose "Game development with
- C++" to create games that use DirectX, Unreal, and Cocos2d. Choose "Linux
- development with C++" to target Linux platforms, including IoT development.
- The **Installation details** pane lists the included and optional components installed
- by each workload. You can select or deselect optional components in this list. For
- example, to support development by using the Visual Studio 2017 or 2015
- compiler toolsets, choose the MSVC v141 or MSVC v140 optional components. You
- can add support for MFC, the experimental Modules language extension,
- IncrediBuild, and more.
- 2. After you choose the workload(s) and optional components you want, choose
- **Install**.
- Next, status screens appear that show the progress of your Visual Studio
- installation.
- **Tip**
- At any time after installation, you can install workloads or components that you
- didn't install initially. If you have Visual Studio open, go to **Tools** > **Get Tools andFeatures...** which opens the Visual Studio Installer. Or, open **Visual Studio Installer**
- from the Start menu. From there, you can choose the workloads or components
- that you wish to install. Then, choose **Modify**.
- # Step 5 - Choose individual components (Optional)
- If you don't want to use the Workloads feature to customize your Visual Studio

- installation, or you want to add more components than a workload installs, you can do
- so by installing or adding individual components from the **Individual components** tab.
- Choose what you want, and then follow the prompts.

Step 6 - Install language packs (Optional)

- By default, the installer program tries to match the language of the operating system
- when it runs for the first time. To install Visual Studio in a language of your choosing,
- choose the **Language packs** tab from the Visual Studio Installer, and then follow the
- prompts.

Change the installer language from the command line

- Another way that you can change the default language is by running the installer from
- the command line. For example, you can force the installer to run in English by using the
- following command: `vs_installer.exe --locale en-US` . The installer will remember this
- setting when it's run the next time. The installer supports the following language tokens:
- zh-cn, zh-tw, cs-cz, en-us, es-es, fr-fr, de-de, it-it, ja-jp, ko-kr, pl-pl, pt-br, ru-ru, and tr-tr.

Step 7 - Change the installation location (Optional)

- You can reduce the installation footprint of Visual Studio on your system drive. You can
- choose to move the download cache, shared components, SDKs, and tools to different
- drives, and keep Visual Studio on the drive that runs it the fastest.
- **Important**
- You can select a different drive only when you first install Visual Studio. If you've
- already installed it and want to change drives, you must uninstall Visual Studio and
- then reinstall it.

Step 8 - Start developing

- 1. After Visual Studio installation is complete, choose the **Launch** button to get
- started developing with Visual Studio.
- 2. On the start window, choose **Create a new project**.

- 3. In the search box, enter the type of app you want to create to see a list of available
- templates. The list of templates depends on the workload(s) that you chose during
- installation. To see different templates, choose different workloads.
- You can also filter your search for a specific programming language by using the
- **Language** drop-down list. You can filter by using the **Platform** list and the **Project**
- **type** list, too.
- 4. Visual Studio opens your new project, and you're ready to code!
- When Visual Studio is running, you're ready to continue to the next step.

Next Steps

What is Visual Studio?

- Article • 06/19/2024
- Visual Studio is a powerful developer tool that you can use to complete the entire
- development cycle in one place. It's a comprehensive integrated development
- environment (IDE) that you can use to write, edit, debug, and build code. Then deploy
- your app. Visual Studio includes compilers, code completion tools, source control,
- extensions, and many other features to enhance every stage of the software
- development process.
- With the variety of features and languages support in Visual Studio, you can grow from
- writing your first "Hello World" program to developing and deploying apps. For
- example, build, debug, and test .NET and C++ apps, edit ASP.NET pages in the web
- designer view, develop cross-platform mobile and desktop apps with .NET, or build
- responsive Web UIs in C#.
- To install Visual Studio, select the following button, and choose the edition of Visual
- Studio to download.

Why use Visual Studio?

- Visual Studio provides developers a feature rich development environment to develop
- high-quality code efficiently and collaboratively.
- Workload-based installer - install only what you need
- Powerful coding tools and features - everything you need to build your apps in
- one placeMultiple language support - code in C++, C#, JavaScript, TypeScript, Python, and
- more
- Cross-platform development - build apps for any platform
- Version control integration - collaborate on code with team mates
- AI-assisted development - write code more efficiently with AI assistance

Discover Visual Studio

- Visual Studio supports different parts of the software development cycle.

Develop your code

- Visual Studio IDE provides many features that make it easier for you to write and
- manage your code with confidence. For example, code quickly and accurately with AI
- assisted development tools. These tools include GitHub Copilot and IntelliCode. Make
- quick improvements to your code using light bulbs that suggest actions, or
- expand/collapse blocks of code using outlining. Organize and explore your code with
- the **Solution Explorer** that shows your code organized by files or the **Class View** that
- shows your code organized by classes.
- Learn more about all the features in the IDE that help you organize and edit content:
- Code editor
- Personalize the IDE and the editor
- Organize code
- Tips and tricks

AI-assisted development

- GitHub Copilot, GitHub Copilot Chat, and IntelliCode assist developers in writing code
- faster and with greater accuracy, help develop a deeper understanding of the codebase,
- and help with other development tasks such as writing unit tests, debugging, and
- profiling.

- Learn more about AI-assisted development in Visual Studio:
- Get started with GitHub Copilot in Visual Studio:
- Install and manage GitHub Copilot
- Use GitHub Copilot Completions in Visual Studio
- Use GitHub Copilot Chat in Visual Studio
- Debug with Copilot## Build your app
- You can compile and build your applications to create builds immediately and test them
- in a debugger. You can run multi-processor builds for C++ and C# projects. Visual
- Studio also provides several options that you can configure when you build applications.
- You can create a custom build configuration in addition to the built-in configurations,
- hide certain warning messages, or increase build output information.
- Learn more about how to compile and build in Visual Studio:
- Create build configurations for your project
- Build an application
- ## Debug your code
- Integrated debugging in Visual Studio enables you to debug, profile, and diagnose with
- ease. You step through your code and look at the values stored in variables, set watches
- on variables to see when values changes, examine the execution path of your code.
- Visual Studio offers other ways to debug your code while it runs.
- Learn more about debugging effectively in Visual Studio:
- Debug your app
- Debugging techniques and tools
- Measure app performance
- Debug with Copilot
- Tips and tricks
- ## Test your code
- You can write high-quality code with comprehensive testing tools in Visual Studio. Unit
- tests give developers and testers a quick way to find logic errors in code. You can
- analyze how much code you're testing and see instant results in a test suite. Know the
- impact of every change you make with advanced features that test code while you type.
- Learn more about the testing tools available in Visual Studio:
- Use testing tools in Visual Studio
- Create and run unit tests

- Analyze code coverage
- **Version control**With the integrated Git features in Visual Studio, you can clone, create, or open your
- own repositories. The Git tool window has everything you need to commit and push
- changes, manage branches, and resolve merge conflicts. If you have a GitHub account,
- you can manage those repos directly within Visual Studio.
- Learn more about version control in Visual Studio:
- Version control with Git
- Visual Studio and GitHub
- **Collaborate with others**
- Visual Studio Live Share enables real-time collaborative development. With Live Share
- you can share your project with your peers, no matter the language or platform. Get to
- the bottom of an issue fast by allowing your team to connect, navigate, set break points,
- and type in your editor session.
- Learn more about how to collaborate with Live Share:
- Collaborate with Live Share
- Common use cases
- **Deploy your app**
- By deploying an application, service, or component, you distribute it for installation on
- other computers, devices, or servers, or in the cloud. You can choose the appropriate
- method in Visual Studio for the type of deployment that you need. Share your apps and
- code by publishing to the web or Azure, or by deploying to a network share or a local
- folder.
- Learn more about how to deploy your app using Visual Studio:
- Deploy your app from Visual Studio
- Deploy your app to a folder, a web server, Azure, or another destination
- **Choose your Visual Studio edition**
- There are three editions of Visual Studio:
- Community - free, fully featured IDE for students, open-source developers, and
- individual developers.
- Professional - a subscription based option for individual developers or small

- teams.# Feedback
- **Was this page helpful?**
- Provide product feedback
- | Ask the community
- Enterprise - a subscription based option for small to large business and
- enterprise organizations.
- Compare features across Visual Studio editions and acquire the Visual Studio
- edition that best fits your needs.
- Select the following button to install Visual Studio, and choose the edition of Visual
- Studio.
- Dive into coding with one of the following language-specific tutorials:
- Create a simple C# console app
- Get started with Python
- Create a simple VB console app
- Create a C++ console app
- Create a Node.js and Express app
- To develop any type of app, or learn a language, you work in the feature rich Visual
- Studio Integrated Development Environment (IDE). Explore Visual Studio further with
- one of these introductory articles:
- Tour the IDE to get familiar with the IDE features and to learn how to use it for
- basic tasks.
- Cover the basics in this Learn module: Introduction to Visual Studio
- # Install Visual Studio
-
- # Get started
- # Related content
- **Yes**
- **No**# Create a C++ console app project
- Article • 07/06/2023
- The usual starting point for a C++ programmer is a "Hello, world!" application that runs
- on the command line. That's what you create in Visual Studio in this step.
- # Prerequisites

- Have Visual Studio with the Desktop development with C++ workload installed
- and running on your computer. If it's not installed yet, see Install C++ support in
- Visual Studio.

Create your app project

- Visual Studio uses *projects* to organize the code for an app, and *solutions* to organize
- your projects. A project contains all the options, configurations, and rules used to build
- your apps. It manages the relationship between all the project's files and any external
- files. To create your app, first, create a new project and solution.
- 1. In Visual Studio, open the **File** menu and choose **New > Project** to open the **Create**
- **a new Project** dialog. Select the **Console App** template that has **C++**, **Windows**,
- and **Console** tags, and then choose **Next**.2. In the **Configure your new project** dialog, enter *HelloWorld* in the **Project name**
- edit box. Choose **Create** to create the project.
- Visual Studio creates a new project. It's ready for you to add and edit your source
- code. By default, the Console App template provides source code for a "Hello
- World" app, like this:
- When the code looks like this in the editor, you're ready to go on to the next step
- and build your app.I ran into a problem.

Next steps

-

Troubleshooting guide

- Come here for solutions to common issues when you create your first C++ project.

Create your app project: issues

- The **New Project** dialog should show a **Console App** template that has **C++**, **Windows**,
- and **Console** tags. If you don't see it, there are two possible causes. It might be filtered
- out of the list, or it might not be installed. First, check the filter dropdowns at the top of
- the list of templates. Set them to **C++**, **Windows**, and **Console**. The C++ **Console App**

- template should appear; otherwise, the **Desktop development with C++** workload isn't
- installed.
- To install **Desktop development with C++**, you can run the installer right from the **New**
- **Project** dialog. Choose the **Install more tools and features** link at the bottom of the
- template list to start the installer. If the **User Account Control** dialog requests
- permissions, choose **Yes**. In the installer, make sure the **Desktop development with C++**
- workload is checked. Then choose **Modify** to update your Visual Studio installation.
- If another project with the same name already exists, choose another name for your
- project. Or, delete the existing project and try again. To delete an existing project, delete
- the solution folder (the folder that contains the `helloworld.sln` file) in File Explorer.
- Go back.

Build and run a C++ console app project

- Article • 07/01/2024
- In Create a C++ console app project you created a C++ console app project and
- entered your code. Now you can build and run it within Visual Studio. Then, run it as a
- stand-alone app from the command line.

Prerequisites

- Have Visual Studio with the Desktop development with C++ workload installed
- and running on your computer. If it's not installed, follow the steps in Install C++
- support in Visual Studio.
- Create a "Hello, World!" project. By default, it contains code to print `Hello World!` .
- If you haven't done this step yet, follow the steps in Create a C++ console app
- project.
- If Visual Studio looks like this, you're ready to build and run your app:

- # Build and run your code in Visual Studio1. To build your project, from the main menu choose **Build** > **Build Solution**. The
- **Output** window shows the results of the build process.
- 2. To run the code, on the menu bar, choose **Debug**, **Start without debugging**.
- A console window opens and then runs your app. When you start a console app in
- Visual Studio, it runs your code, then prints "Press any key to continue . . ." to give
- you a chance to see the output.Congratulations! You created your first "Hello, world!" console app in Visual Studio!
- Press a key to dismiss the console window and return to Visual Studio.
- I ran into a problem.
- # Run your code in a command window
- Normally, you run console apps at the command prompt, not in Visual Studio. Once
- Visual Studio builds your app, you can run it from a command window. Here's how to
- find and run your new app in a command prompt window.
- 1. In **Solution Explorer**, select the HelloWorld solution (not the HelloWorld project)
- and right-click to open the context menu. Choose **Open Folder in File Explorer** to
- open a **File Explorer** window in the HelloWorld solution folder.
- 2. In the **File Explorer** window, open the `x64` folder and then the `Debug` folder. This
- folder contains your app, `HelloWorld.exe` , and debugging files. Hold down the
- **Shift** key and right-click on `HelloWorld.exe` to open the context menu. Choose
- **Copy as path** to copy the path to your app to the clipboard. If you see
- `HelloWorld.exe.recipe` , it's because you did the **Open Folder in File Explorer** step
- on the HelloWorld *project* instead of the HelloWorld *solution*. Navigate up a level in
- File Explorer to get to the solution folder. This folder also contains a `x64\Debug\`
- folder, where `HelloWorld.exe` is.
- 3. To open a command prompt window, press **Windows+R** to open the **Run** dialog.

- Enter *cmd.exe* in the **Open** textbox, then choose **OK** to run a command prompt
- window.
- 4. In the command prompt window, right-click to paste the path to your app into the
- command prompt. Press Enter to run your app.Congratulations, you built and ran a console app in Visual Studio!
- I ran into a problem.

Next Steps

- Once you build and run this simple app, you're ready for more complex projects. For
- more information, see Using the Visual Studio IDE for C++ Desktop Development. It has
- more detailed walkthroughs that explore the capabilities of Microsoft C++ in Visual
- Studio.

Troubleshooting guide

- Come here for solutions to common issues when you create your first C++ project.

Build and run your code in Visual Studio: issues

- If red squiggles appear under anything in the source code editor, the build may have
- errors or warnings. Check that your code matches the example in spelling, punctuation,
- and case.
- Go back.

Run your code in a command window: issuesFeedback

- **Was this page helpful?**
- Provide product feedback
- | Get help at Microsoft Q&A
- If the path shown in File Explorer ends in `\HelloWorld\HelloWorld` , you opened the
- HelloWorld *project* instead of the HelloWorld *solution*. You won't see your app in the
- `x64\Debug` folder. Navigate up a level in File Explorer to get to the solution folder, the
- first `HelloWorld` in the path. This folder also contains a `x64\Debug` folder, where your
- app is.

- You can also navigate to the solution `x64\Debug` folder at the command line to run your
- app. Your app won't run from other directories without specifying the path to the app.
- However, you can copy your app to another directory and run it from there. It's also
- possible to copy it to a directory specified by your `PATH` environment variable, then run
- it from anywhere.
- If you don't see **Copy as path** in the shortcut menu, dismiss the menu, and then hold
- down the **Shift** key while you open it again. This command is just for convenience. You
- can also copy the path to the folder from the File Explorer search bar, and paste it into
- the **Run** dialog, and then enter the name of your executable at the end. It's just a little
- more typing, but it has the same result.
- Go back.
- **Yes**
- **No**

Welcome back to C++ - Modern C++

- Article • 11/07/2022
- Since its creation, C++ has become one of the most widely used programming
- languages in the world. Well-written C++ programs are fast and efficient. The language
- is more flexible than other languages: It can work at the highest levels of abstraction,
- and down at the level of the silicon. C++ supplies highly optimized standard libraries. It
- enables access to low-level hardware features, to maximize speed and minimize memory
- requirements. C++ can create almost any kind of program: Games, device drivers, HPC,
- cloud, desktop, embedded, and mobile apps, and much more. Even libraries and
- compilers for other programming languages get written in C++.
- One of the original requirements for C++ was backward compatibility with the C
- language. As a result, C++ has always permitted C-style programming, with raw
- pointers, arrays, null-terminated character strings, and other features. They may enable

- great performance, but can also spawn bugs and complexity. The evolution of C++ has
- emphasized features that greatly reduce the need to use C-style idioms. The old C
- programming facilities are still there when you need them. However, in modern C++
- code you should need them less and less. Modern C++ code is simpler, safer, more
- elegant, and still as fast as ever.
- The following sections provide an overview of the main features of modern C++. Unless
- noted otherwise, the features listed here are available in C++11 and later. In the
- Microsoft C++ compiler, you can set the /std compiler option to specify which version
- of the standard to use for your project.

Resources and smart pointers

- One of the major classes of bugs in C-style programming is the *memory leak*. Leaks are
- often caused by a failure to call **delete** for memory that was allocated with **new** .
- Modern C++ emphasizes the principle of *resource acquisition is initialization* (RAII). The
- idea is simple. Resources (heap memory, file handles, sockets, and so on) should be
- *owned* by an object. That object creates, or receives, the newly allocated resource in its
- constructor, and deletes it in its destructor. The principle of RAII guarantees that all
- resources get properly returned to the operating system when the owning object goes
- out of scope.
- To support easy adoption of RAII principles, the C++ Standard Library provides three
- smart pointer types: std::unique_ptr, std::shared_ptr, and std::weak_ptr. A smart pointer
- handles the allocation and deletion of the memory it owns. The following exampleshows a class with an array member that is allocated on the heap in the call to
- `make_unique()` . The calls to **new** and **delete** are encapsulated by the `unique_ptr` class.
- When a `widget` object goes out of scope, the unique_ptr destructor will be invoked and
- it will release the memory that was allocated for the array.
- `C++`

- Whenever possible, use a smart pointer to manage heap memory. If you must use the
- **new** and **delete** operators explicitly, follow the principle of RAII. For more information,
- see Object lifetime and resource management (RAII).
- C-style strings are another major source of bugs. By using std::string and std::wstring,
- you can eliminate virtually all the errors associated with C-style strings. You also gain the
- benefit of member functions for searching, appending, prepending, and so on. Both are
- highly optimized for speed. When passing a string to a function that requires only read
- only access, in C++17 you can use std::string_view for even greater performance benefit.
- The standard library containers all follow the principle of RAII. They provide iterators for
- safe traversal of elements. And, they're highly optimized for performance and have been
- thoroughly tested for correctness. By using these containers, you eliminate the potential
- `#include <memory>`
- `class widget`
- `{`
- `private:`
- `std::unique_ptr<int[]> data;`
- `public:`
- `widget(const int size) { data = std::make_unique<int[]>(size); }`
- `void do_something() {}`
- `};`
- `void functionUsingWidget() {`
- `widget w(1000000); // lifetime automatically tied to enclosing scope`
- `// constructs w, including the w.data gadget member`
- `// ...`
- `w.do_something();`
- `// ...`
- `} // automatic destruction and deallocation for w and w.data`
- **std::string and std::string_view**
- **std::vector and other Standard Library containers**for bugs or inefficiencies that might be introduced in custom data structures. Instead of
- raw arrays, use vector as a sequential container in C++.
- C++
- `vector<string> apples;`

- `apples.push_back("Granny Smith");`
- Use map (not `unordered_map`) as the default associative container. Use set, multimap,
- and multiset for degenerate and multi cases.
- C++
- `map<string, string> apple_color;`
- `// ...`
- `apple_color["Granny Smith"] = "Green";`
- When performance optimization is needed, consider using:
- Unordered associative containers such as unordered_map. These have lower per
- element overhead and constant-time lookup, but they can be harder to use
- correctly and efficiently.
- Sorted `vector` . For more information, see Algorithms.
- Don't use C-style arrays. For older APIs that need direct access to the data, use accessor
- methods such as `f(vec.data(), vec.size());` instead. For more information about
- containers, see C++ Standard Library Containers.
- # Standard Library algorithms
- Before you assume that you need to write a custom algorithm for your program, first
- review the C++ Standard Library algorithms. The Standard Library contains an ever
- growing assortment of algorithms for many common operations such as searching,
- sorting, filtering, and randomizing. The math library is extensive. In C++17 and later,
- parallel versions of many algorithms are provided.
- Here are some important examples:
- `for_each` , the default traversal algorithm (along with range-based `for` loops).
- `transform` , for not-in-place modification of container elements
- `find_if` , the default search algorithm.
- `sort` , `lower_bound` , and the other default sorting and searching algorithms.To write a comparator, use strict < and use *named lambdas* when you can.
- C++
- C++11 introduced the auto keyword for use in variable, function, and template
- declarations. **auto** tells the compiler to deduce the type of the object so that you don't
- have to type it explicitly. **auto** is especially useful when the deduced type is a nested
- template:

- C++
- C-style iteration over arrays and containers is prone to indexing errors and is also
- tedious to type. To eliminate these errors, and make your code more readable, use
- range-based **for** loops with both Standard Library containers and raw arrays. For more
- information, see Range-based for statement.
- C++
- `auto comp = [](const widget& w1, const widget& w2)`
- `{ return w1.weight() < w2.weight(); }`
- `sort( v.begin(), v.end(), comp );`
- `auto i = lower_bound( v.begin(), v.end(), widget{0}, comp );`
- # auto instead of explicit type names
- `map<int,list<string>>::iterator i = m.begin(); // C-style`
- `auto i = m.begin(); // modern C++`
- # Range-based for loops
- `#include <iostream>`
- `#include <vector>`
- `int main()`
- `{`
- `std::vector<int> v {1,2,3};`
- `// C-style`
- `for(int i = 0; i < v.size(); ++i)`
- `{`
- `std::cout << v[i];`
- }Macros in C and C++ are tokens that are processed by the preprocessor before
- compilation. Each instance of a macro token is replaced with its defined value or
- expression before the file is compiled. Macros are commonly used in C-style
- programming to define compile-time constant values. However, macros are error-prone
- and difficult to debug. In modern C++, you should prefer constexpr variables for
- compile-time constants:
- C++
- In modern C++, you can use brace initialization for any type. This form of initialization is
- especially convenient when initializing arrays, vectors, or other containers. In the
- following example, `v2` is initialized with three instances of `S` . `v3` is initialized with three
- instances of `S` that are themselves initialized using braces. The compiler infers the type
- of each element based on the declared type of `v3` .

- C++
- // Modern C++:
- for(auto& num : v)
- {
- std::cout << num;
- }
- }

- # constexpr expressions instead of macros

- #define SIZE 10 // C-style
- constexpr int size = 10; // modern C++

- # Uniform initialization

- #include <vector>
- struct S
- {
- std::string name;
- float num;
- S(std::string s, float f) : name(s), num(f) {}
- };
- int main()
- {
- // C-style initialization
- std::vector<S> v;
- S s1("Norah", 2.7);For more information, see Brace initialization.
- Modern C++ provides *move semantics*, which make it possible to eliminate unnecessary
- memory copies. In earlier versions of the language, copies were unavoidable in certain
- situations. A *move* operation transfers ownership of a resource from one object to the
- next without making a copy. Some classes own resources such as heap memory, file
- handles, and so on. When you implement a resource-owning class, you can define a
- *move constructor* and *move assignment operator* for it. The compiler chooses these
- special members during overload resolution in situations where a copy isn't needed. The
- Standard Library container types invoke the move constructor on objects if one is
- defined. For more information, see Move Constructors and Move Assignment Operators
- (C++).
- In C-style programming, a function can be passed to another function by using a

- *function pointer*. Function pointers are inconvenient to maintain and understand. The
- function they refer to may be defined elsewhere in the source code, far away from the
- point at which it's invoked. Also, they're not type-safe. Modern C++ provides *function*
- *objects*, classes that override the operator() operator, which enables them to be called
- like a function. The most convenient way to create function objects is with inline lambda
- expressions. The following example shows how to use a lambda expression to pass a
- function object, that the `find_if` function will invoke on each element in the vector:
- C++
- `S s2("Frank", 3.5);`
- `S s3("Jeri", 85.9);`
- `v.push_back(s1);`
- `v.push_back(s2);`
- `v.push_back(s3);`
- `// Modern C++:`
- `std::vector<S> v2 {s1, s2, s3};`
- `// or...`
- `std::vector<S> v3{ {"Norah", 2.7}, {"Frank", 3.5}, {"Jeri", 85.9} };`
- `}`
- # Move semantics
- # Lambda expressionsThe lambda expression `[=] (int i) { return i > x && i < y; }` can be read as "function
- that takes a single argument of type **int** and returns a boolean that indicates whether
- the argument is greater than `x` and less than `y` ." Notice that the variables `x` and `y`
- from the surrounding context can be used in the lambda. The `[=]` specifies that those
- variables are *captured* by value; in other words, the lambda expression has its own
- copies of those values.
- Modern C++ emphasizes exceptions, not error codes, as the best way to report and
- handle error conditions. For more information, see Modern C++ best practices for
- exceptions and error handling.
- Use the C++ Standard Library std::atomic struct and related types for inter-thread
- communication mechanisms.

- Unions are commonly used in C-style programming to conserve memory by enabling
- members of different types to occupy the same memory location. However, unions
- aren't type-safe and are prone to programming errors. C++17 introduces the
- std::variant class as a more robust and safe alternative to unions. The std::visit function
- can be used to access the members of a `variant` type in a type-safe manner.
- C++ Language Reference
- Lambda Expressions
- C++ Standard Library
- Microsoft C/C++ language conformance
- `std::vector<int> v {1,2,3,4,5};`
- `int x = 2;`
- `int y = 4;`
- `auto result = find_if(begin(v), end(v), [=](int i) { return i > x && i <`
- `y; });`
- # Exceptions
- # std::atomic
- # std::variant (C++17)
- # See alsoFeedback
- **Was this page helpful?**
- Provide product feedback
- | Get help at Microsoft Q&A
- **Yes**
- **No**# Create a console calculator in C++
- Article • 10/08/2024
- The usual starting point for a C++ programmer is a "Hello, world!" application that runs
- on the command line. You start with that in this article, and then move on to something
- more challenging: a calculator app.
- # Prerequisites
- Visual Studio with the **Desktop development with C++** workload installed and
- running on your computer. To install it, see Install C++ support in Visual Studio.

- This tutorial demonstrates a feature called edit and continue which allows you to
- make changes to your code while the app is running. To enable edit and continue,
- from the main menu select **Tools** > **Options** > **Debugging** > **General** and ensure
- that **Require source files to exactly match the original version** is checked.

Create your app project

- Visual Studio uses *projects* to organize the code for an app, and *solutions* to organize
- one or more projects. A project contains all the options, configurations, and rules used
- to build an app. It also manages the relationship between all the project's files and any
- external files. To create your app, first, create a new project and solution.
- 1. Start Visual Studio--the Visual Studio Start dialog box appears. Select **Create a new**
- **project** to get started.2. In the **Create a new project** dialog, set the language dropdown to **C++**, set the
- platform dropdown to **Windows**, select **Console App** from the list of project types,
- then select **Next**.
- **Important**
- Make sure you select the C++ version of the **Console App** template. It has the
- **C++**, **Windows**, and **Console** tags, and the icon has "++" in the corner.3. In the **Configure your new project** dialog box, select the **Project name** text box,
- name your new project *CalculatorTutorial*, then select **Create**.
- An empty C++ Windows console application 'Hello World' app is created. Console
- applications use a Windows console window to display output and accept user
- input. In Visual Studio, an editor window opens and shows the generated code:
- C++
- ```
  // CalculatorTutorial.cpp : This file contains the 'main' function.
  Program execution begins and ends there.
  //
  #include <iostream>
  int main()
  {
  std::cout << "Hello World!\n";
  }
  ```
  ```

- `// Run program: Ctrl + F5 or Debug > Start Without Debugging menu`
- `// Debug program: F5 or Debug > Start Debugging menu`
- `// Tips for Getting Started:`
- `// 1. Use the Solution Explorer window to add/manage files`
- `// 2. Use the Team Explorer window to connect to source control`
- `// 3. Use the Output window to see build output and other messages`
- `// 4. Use the Error List window to view errors`
- `// 5. Go to Project > Add New Item to create new code files, or`
- `Project > Add Existing Item to add existing code files to the project// 6. In the future, to open this project again, go to File > Open >`
- `Project and select the .sln file`
- # Verify that your new app builds and runs
- The template for a new Windows console application creates a simple C++ "Hello
- World" app. At this point, you can see how Visual Studio builds and runs the apps you
- create right from the IDE.
- 1. To build your project, select **Build Solution** from the **Build** menu. The **Output**
- window shows the results of the build process.
- 2. To run the code, on the menu bar, select **Debug** > **Start without debugging**
- (Ctrl+F5).A console window opens and your app runs within it.
- When you start a console app in Visual Studio, it runs your code, then prints "Press
- any key to close this window . . ." to give you a chance to see the output.
- Congratulations! You created your first "Hello, world!" console app in Visual Studio!
- 3. Press a key to dismiss the console window and return to Visual Studio.
- You now have the tools to build and run your app after every change, to verify that the
- code still works as you expect. Later, we show you how to debug it if it doesn't.
- Now let's modify the code in this template to be a calculator app.
- 1. Replace the contents of the *CalculatorTutorial.cpp* file with the following code so
- that it matches this example:
- C++
- # Edit the code
- `// CalculatorTutorial.cpp : This file contains the 'main' function.`
- `Program execution begins and ends there.`
- `//`

- `#include <iostream>`
- `using namespace std;`
- `int main()`
- `{`
- `cout << "Calculator Console Application" << endl << endl;`
- `cout << "Please enter the operation to perform. Format: a+b | a-b |`
  Understanding the code:
- The `#include` statement brings in code in other files. Sometimes, you
- may see a filename surrounded by angle brackets like `<iostream>` . The
- angle brackets instruct the compiler to look for the `iostream` header file
- first in the standard system directories, and if not found, to look in
- directories specific to the project. Other times, you may see a filename
- surrounded by quotes like `"someHeader.h"` . The quotes instruct the
- compiler to skip looking in the standard system directories and instead
- only look in directories specific to the project.
- The `using namespace std;` tells the compiler to expect code from the
- C++ Standard Library to be used in this file. Without this line, each
- keyword from the library would have to be preceded with `std::` to
- denote its scope. For instance, without that line, each reference to `cout`
- would be written as `std::cout` . The **using** statement is added to make
  it
- more convenient to access code in another namespace.
- The `cout` keyword is used to print to standard output in C++. The `<<`
- operator tells the compiler to send whatever is to the right of it to the
- standard output.
- The `endl` keyword is like the Enter key; it ends the line and moves the
- cursor to the next line. It's a better practice to put a `\n` inside the string
- (contained by `""`) to do the same thing because `endl` always flushes
  the
- buffer which can hurt the performance of the program. But since this is
  a
- very small app, `endl` is used instead.
- All C++ statements must end with semicolons and all C++
  applications
- must contain a `main()` function. This function is what the program runs
  at
- `a*b | a/b"`
- `<< endl;`
- `return 0;`
- `}`
- `// Run program: Ctrl + F5 or Debug > Start Without Debugging menu`
- `// Debug program: F5 or Debug > Start Debugging menu`
- `// Tips for Getting Started:`
- `// 1. Use the Solution Explorer window to add/manage files`
- `// 2. Use the Team Explorer window to connect to source control`
- `// 3. Use the Output window to see build output and other messages`
- `// 4. Use the Error List window to view errors`
- `// 5. Go to Project > Add New Item to create new code files, or`

- `Project > Add Existing Item to add existing code files to the project`
- `// 6. In the future, to open this project again, go to File > Open >`
- `Project and select the .sln file`the start. All code must be accessible from `main()` in order to be used.
- 2. To save the file, press **Ctrl+S**, or select the floppy disk icon in the toolbar under the
- menu bar.
- 3. To run the application, press **Ctrl+F5** or go to the **Debug** menu and select **Start**
- **Without Debugging**. You should see a console window appear that looks like this.
- 4. Close the console window when you're done.
- # Add code to do some math
- A class is like a blueprint for an object that does something. In this case, we define a
- calculator class to contain the math logic.
- # Add a Calculator class
- 1. Go to the **Project** menu and select **Add Class**. In the **Class Name** edit box, enter
- *Calculator*. Select **OK**.
- Two new files get added to your project. To save all your changed files at once,
- press **Ctrl+Shift+S**. It's a keyboard shortcut for **File** > **Save All**. There's also a
- toolbar button for **Save All**, an icon of two floppy disks, found beside the **Save**button. In general, it's good practice to do **Save All** frequently, so you don't miss
- saving any changes.
- The **Add Class** wizard creates `.h` and `.cpp` files that have the same name as the
- class. You can see a full list of your project files in the **Solution Explorer** window,
- visible on the side of the IDE. If the window isn't visible, open it from the menu bar
- via **View** > **Solution Explorer**.
- You can open a file by double-clicking it in the **Solution Explorer** window. Double
- click `Calculator.h` to open it.
- 2. Replace the contents of **Calculator.h** with the following code so that the file now
- looks like this:
- C++
- Understanding the code

- This code declares a new function called `Calculate` , which handles math
- operations for addition, subtraction, multiplication, and division.
- `#pragma once`
- `class Calculator`
- `{`
- `public:`
- `double Calculate(double x, char oper, double y);`
- `};`C++ code is organized into *header* (`.h`) files and *source* (`.cpp`) files.
- Some other file extensions are supported by various compilers, but these
- are the main ones to know about. Functions and variables are normally
- *declared*, that is, given a name and a type, in header files, and
- *implemented*, or given a definition, in source files. To access code defined
- in another file, you can use `#include "filename.h"` , where `filename.h` is
- the name of the file that declares the variables or functions you want to
- use.
- It's good practice to organize your code into different files based on what
- it does, so it's easy to find the code you need later. In our case, we define
- the `Calculator` class separately from the file containing the `main()`
- function, but we plan to reference the `Calculator` class in `main()` .
- 3. A green squiggle appears under `Calculate` because although the `Calculate`
- function is *declared*, it isn't *defined*. Hover over `Calculate` , click the down arrow on
- the screwdriver icon, and select **Create definition of 'Calculate' in Calculator.cpp** .
- This code is added to *Calculator.cpp* :Currently, it just returns 0.0. Let's change that.
- 4. Switch to the *Calculator.cpp* file in the editor window. Replace the contents of
- *Calculator::Calculate(double x, char oper, double y)* with:
- `C++`
- Understanding the code
- The function `Calculate` takes a number, an operator, and a second
- number. Then it performs the requested operation on the two numbers.
- The `switch` statement checks which operator was provided, and executes
- the case corresponding to that operation. The `default:` case is a fallback
- `double Calculator::Calculate(double x, char oper, double y)`

- {
- switch(oper)
- {
- case '+':
- return x + y;
- case '-':
- return x - y;
- case '*':
- return x * y;
- case '/':
- return x / y;
- default:
- return 0.0;
- }
- }in case the user types an operator that isn't handled by any of the
- preceding `case` statements. It's best to handle invalid user input in a
- more elegant way, but this is beyond the scope of this tutorial.
- The **double** keyword denotes a type of number that supports decimals.
- This type of number is called a floating-point number, and `double` means
- a floating point number that has extra precision. This way, the calculator
- can handle both decimal math and integer math. The `Calculate` function
- is required to always return a double-precision floating point number due
- to the **double** at the start of the code (this denotes the function's return
- type), which is why we return 0.0 in the default case.
- The `.h` file declares the function *prototype*, which tells the compiler
- upfront what parameters it requires, and what return type to expect from
- it. The `.cpp` file has all the implementation details of the function.
- If you build and run the code again at this point, it immediately exits after asking which
- operation to perform. So, modify the `main` function to do multiple calculations.
- 1. Update the `main` function in *CalculatorTutorial.cpp* as follows:
- C++
- # Call the Calculator class member functions
- // CalculatorTutorial.cpp : This file contains the 'main' function.
- Program execution begins and ends there.
- //
- #include <iostream>
- #include "Calculator.h"
- using namespace std;
- int main()
- {
- double x = 0.0;
- double y = 0.0;

- `double result = 0.0;`
- `char oper = '+';`
- `cout << "Calculator Console Application" << endl << endl;`
- `cout << "Please enter the operation to perform. Format: a+b | a-b |`
- `a*b | a/b"`
- `<< endl;`
- `Calculator c;`
- `while (true)`
- {Understanding the code
- Since C++ programs always start at the `main()` function, we need to call
- our other code from there, so an `#include` statement is needed to make
- that code visible to our `main()` function.
- The variables `x` , `y` , `oper` , and `result` are declared to store the first
- number, second number, operator, and final result, respectively. It's
- always good practice to give them some initial values to avoid undefined
- behavior, which is what is done here.
- The `Calculator c;` line declares an object named `c` as an instance of the
- `Calculator` class. The class itself is just a blueprint for how calculators
- work; the object is the specific calculator that does the math.
- The `while (true)` statement is a loop. The code inside the loop executes
- over and over again as long as the condition inside the `()` holds true.
- Since the condition is simply listed as **true** , it's always true, so the loop
- runs forever. To close the program, the user must manually close the
- console window. Otherwise, the program always waits for new input.
- The `cin` keyword accepts input from the user. The input stream is smart
- enough to process a line of text entered in the console window and place
- it inside each of the variables listed, in order.
- The `c.Calculate(x, oper, y);` expression calls the `Calculate` function
- defined earlier, and supplies the entered input values and the requested
- operation. The function then returns a number that is stored in `result` .
- Finally, `result` is printed to the console and the user sees the result of
- the calculation.
- Now test the program again to make sure everything works properly.
- 1. Press **Ctrl+F5** to rebuild and start the app.
- 2. Enter `5+5` , and press **Enter**. Verify that the result is 10.
- `cin >> x >> oper >> y;`
- `result = c.Calculate(x, oper, y);`
- `cout << "Result " << "of " << x << oper << y << " is: " <<`
- `result << endl;`
- }
- `return 0;`
- }

- ## Build and test the code again3. Stop the program by closing the console window.
- # Debug the app
- Since the user is free to type anything into the console window, let's make sure the
- calculator handles unexpected input. Instead of running the program, let's debug it so
- we can inspect what it's doing step-by-step.
- # Run the app in the debugger
- 1. In `CalcuatorTutorial.cpp` , set a breakpoint on the line: `result = c.Calculate(x,`
- `oper, y);` . To set the breakpoint, click next to the line in the gray vertical bar along
- the left edge of the editor window so that a red dot appears.
- Now when we debug the program, execution pauses at that line. We already have
- a rough idea that the program works for simple cases. Since we don't want to
- pause execution every time we call `Calculate()` , let's make the breakpoint
- conditional.2. Right-click the red dot that represents the breakpoint, and select **Conditions**. In
- the edit box for the condition, enter `(y == 0) && (oper == '/')` . Select the **Close**
- button to save the breakpoint condition.
- Now, execution pauses at the breakpoint when the app tries to divide by 0.
- 3. To debug the program, press **F5**, or select the **Local Windows Debugger** debugger
- toolbar button that has the green arrow icon. In your console app, if you enter
- something like "5 - 0", the program behaves normally and keeps running.
- However, if you type "10 / 0", it pauses at the breakpoint. You can put any number
- of spaces between the operator and numbers: `cin` is smart enough to parse the
- input appropriately.## Useful windows in the debugger
- When you debug your code, you may notice that some new windows appear. These
- windows can assist your debugging experience. Take a look at the **Autos** window. The

- **Autos** window shows you the current values of variables used at least three lines before
- and up to the current line. If you don't see the **Autos** window, from the main menu
- select **Debug** > **Windows** > **Autos**.
- To see all of the variables from that function, switch to the **Locals** window. Because this
- is a small function, the Autos and Locals window show the same variables. But you can
- modify the values of these variables in the Locals window while debugging to see what
- effect they would have on the program. In this case, we leave them alone. Open the
- **Locals** window by selecting **Locals** at the bottom of the **Autos** window, or by selecting
- from the main menu **Debug** > **Windows** > **Locals**.You can also hover over variables in the code to see their current values at the point
- where execution is currently paused. Make sure the editor window is in focus by clicking
- on it first.
- # Continue debugging
- 1. The yellow arrow on the left shows the current point of execution. The current line
- calls `Calculate` , so press **F11** to **Step Into** the function. Now you're executing code
- in the body of the `Calculate` function. Be careful with **Step Into** because it steps
- into any functions on the line you're on, including standard library functions. It's
- fine to step into the standard library, but you may be more interested in focusing
- on your code instead of library code.
- 2. Now that the point of execution is at the start of the `Calculate` function, press **F10**
- to move to the next line in the program's execution. **F10** is also known as **Step**
- **Over**. You can use **Step Over** to move from line to line, without delving into the
- details of what is occurring in each part of the line. In general, you should use **Step**
- **Over** instead of **Step Into** unless you want to dive more deeply into code that is
- being called from elsewhere (as you did to reach the body of `Calculate`).

- 3. Continue using **F10** to **Step Over** each line until you get back to the `main()`
- function in the other file, and stop on the `cout` line.The program is doing what's expected: it takes the first number, and divides it by
- the second. On the `cout` line, hover over the `result` variable or take a look at
- `result` in the **Autos** window. Its value is `inf` , which doesn't look right.
- Let's fix it. The `cout` line outputs whatever value is stored in `result` , so when you
- step one more line forward using **F10**, the console window displays:
- This result is because division by zero is undefined, so the program doesn't have a
- numerical answer for the requested operation.
- # Fix the "divide by zero" error
- Let's handle division by zero more gracefully so that it's easier for the user to
- understand the problem.
- 1. Make the following changes in *CalculatorTutorial.cpp* . You can leave the
- program running as you edit, thanks to a debugger feature called **Edit andContinue**. Add an `if` statement following `cin >> x >> oper >> y;` to check for
- division by zero and output a message to the user if it happens. Otherwise, the
- result is printed.
- `C++`
- 2. Press **F5** once. Program execution continues until it has to pause to ask for user
- input. Enter `10 / 0` again. Now, a more helpful message is printed. The user is
- asked for more input, and the program continues executing normally.
- `// CalculatorTutorial.cpp : This file contains the 'main' function.`
- `Program execution begins and ends there.`
- `//`
- `#include <iostream>`
- `#include "Calculator.h"`
- `using namespace std;`
- `int main()`
- `{`
- `double x = 0.0;`
- `double y = 0.0;`
- `double result = 0.0;`
- `char oper = '+';`
- `cout << "Calculator Console Application" << endl << endl;`
- `cout << "Please enter the operation to perform. Format: a+b | a-b |`
- `a*b | a/b" << endl;`
- `Calculator c;`
- `while (true)`
- `{`

```
cin >> x >> oper >> y;
if (oper == '/' && y == 0)
{
cout << "Math error: Attempted to divide by zero!" << endl;
continue;
}
else
{
result = c.Calculate(x, oper, y);
}
cout << "Result " << "of " << x << oper << y << " is: " <<
result << endl;
}
return 0;
}
```

Note

When you edit code while in debugging mode, there's a risk of code becoming stale. This happens when the debugger is still running your old code, and has not yet updated it with your changes. The debugger displays a dialog to inform you when this happens. Sometimes, you may need to press **F5** to refresh the code being executed. In particular, if you make a change inside a function while the point of execution is inside that function, you need to step out of the function, then back into it again to get the updated code. If that doesn't work and you see an error message, you can stop debugging by clicking on the red square in the toolbar under the menus at the top of the IDE, then start debugging again by entering **F5** or by choosing the green "play" arrow beside the stop button on the toolbar.

Another reason edit and continue may fail is if you see a message that says "The Require source files to exactly match the original version setting under Debug->Options->General needs to be enabled..." To fix this, from the main menu select **Tools** > **Options** > **Debugging** > **General** and ensure that **Require source files to exactly match the original version** is checked.

Understanding the Run and Debug shortcuts

F5, or **Debug** > **Start Debugging**, starts a debugging session, if one isn't

- already active, and runs the program until a breakpoint is hit or the
- program needs user input. If no user input is needed and no breakpoint
- is available to hit, the program terminates and the console window closes
- itself when the program finishes running. If your program outputs to the
- console, use **Ctrl+F5** or set a breakpoint before you press **F5** to keep the
- window open.
- **Ctrl+F5**, or **Debug** > **Start Without Debugging**, runs the application
- without going into debug mode. This is slightly faster than debugging,
- and the console window stays open after the program finishes executing.
- **F10**, known as **Step Over**, lets you iterate through code, line-by-line, and
- visualize how the code is run and what variable values are at each step of
- execution.**F11**, known as **Step Into**, works similarly to **Step Over**, except it steps into
- any functions called on the line of execution. For example, if the line
- being executed calls a function, pressing **F11** moves the pointer into the
- body of the function, so you can follow the function's code being run
- before coming back to the line you started at. Pressing **F10** steps over the
- function call and just moves to the next line; the function call still
- happens, but the program doesn't pause to show you what it's doing.

Close the app

- If it's still running, close the console window to stop the calculator app.

Add Git source control

- Now that you've created an app, you might want to add it to a Git repository. We've got
- you covered. Visual Studio makes that process easy with Git tools you can use directly
- from the IDE.
- **Tip**
- Git is the most widely used modern version control system, so whether you're a
- professional developer or you're learning how to code, Git can be very useful. If
- you're new to Git, the **https://git-scm.com/** website is a good place to start.
- There, you can find cheat sheets, a popular online book, and Git Basics videos.

- To associate your code with Git, you start by creating a new Git repository where your
- code is located. Here's how:
- 1. In the status bar at the bottom-right corner of Visual Studio, select **Add to Source**
- **Control**, and then select **Git**.
- 2. In the **Create a Git repository** dialog box, sign in to GitHub.The repository name auto-populates based on your folder location. By default,
- your new repository is private, which means you're the only one who can access it.
- **Tip**
- Whether your repository is public or private, it's best to have a remote backup
- of your code stored securely on GitHub. Even if you aren't working with a
- team, a remote repository makes your code available to you from any
- computer.
- 3. Select **Create and Push**.
- After you create your repository, status details appear in the status bar.
- The first icon with the arrows shows how many outgoing/incoming commits are in
- your current branch. You can use this icon to pull any incoming commits or push
- any outgoing commits. You can also choose to view these commits first. To do so,
- select the icon, and then select **View Outgoing/Incoming**.

Feedback

- **Was this page helpful?**
- Provide product feedback
- | Get help at Microsoft Q&A
- The second icon with the pencil shows the number of uncommitted changes to
- your code. You can select this icon to view those changes in the **Git Changes**
- window.
- To learn more about how to use Git with your app, see the Visual Studio version control
- documentation.
- Congratulations! You completed the code for the calculator app, built and debugged it,
- and added it to a repo, all in Visual Studio.
- Learn more about Visual Studio for C++

The finished app

- # Next steps
- Yes
- No# Get started with C++/WinRT
- Article • 02/13/2023
- **Important**
- For info about setting up Visual Studio for C++/WinRT development—including
- installing and using the C++/WinRT Visual Studio Extension (VSIX) and the NuGet
- package (which together provide project template and build support)—see **Visual**
- **Studio support for C++/WinRT**.
- To get you up to speed with using C++/WinRT, this topic walks through a simple code
- example based on a new **Windows Console Application (C++/WinRT)** project. This
- topic also shows how to add C++/WinRT support to a Windows Desktop application
- project.
- **Note**
- While we recommend that you develop with the latest versions of Visual Studio and
- the Windows SDK, if you're using Visual Studio 2017 (version 15.8.0 or later), and
- targeting the Windows SDK version 10.0.17134.0 (Windows 10, version 1803), then
- a newly created C++/WinRT project may fail to compile with the error "*error C3861:*
- *'from_abi': identifier not found*", and with other errors originating in *base.h*. The
- solution is to either target a later (more conformant) version of the Windows SDK,
- or set project property **C/C++** > **Language** > **Conformance mode: No** (also, if
- **/permissive-** appears in project property **C/C++** > **Language** > **Command Line**
- under **Additional Options**, then delete it).
- # A C++/WinRT quick-start
- Create a new **Windows Console Application (C++/WinRT)** project.
- Edit `pch.h` and `main.cpp` to look like this.
- C++/WinRT
- `// pch.h`
- `#pragma once`
- `#include <winrt/Windows.Foundation.Collections.h>`C++/WinRT

- Let's take the short code example above piece by piece, and explain what's going on in
- each part.
- C++/WinRT
- With the default project settings, the included headers come from the Windows SDK,
- inside the folder
- `%WindowsSdkDir%Include<WindowsTargetPlatformVersion>\cppwinrt\winrt .` Visual Studio
- includes that path in its *IncludePath* macro. But there's no strict dependency on the
- `#include <winrt/Windows.Web.Syndication.h>`
- `#include <iostream>`
- `// main.cpp`
- `#include "pch.h"`
- `using namespace winrt;`
- `using namespace Windows::Foundation;`
- `using namespace Windows::Web::Syndication;`
- `int main()`
- `{`
- `winrt::init_apartment();`
- `Uri rssFeedUri{ L"https://blogs.windows.com/feed" };`
- `SyndicationClient syndicationClient;`
- `syndicationClient.SetRequestHeader(L"User-Agent", L"Mozilla/5.0`
- `(compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)");`
- `SyndicationFeed syndicationFeed =`
- `syndicationClient.RetrieveFeedAsync(rssFeedUri).get();`
- `for (const SyndicationItem syndicationItem : syndicationFeed.Items())`
- `{`
- `winrt::hstring titleAsHstring = syndicationItem.Title().Text();`
- `// A workaround to remove the trademark symbol from the title`
- `string, because it causes issues in this case.`
- `std::wstring titleAsStdWstring{ titleAsHstring.c_str() };`
- `titleAsStdWstring.erase(remove(titleAsStdWstring.begin(),`
- `titleAsStdWstring.end(), L'™'), titleAsStdWstring.end());`
- `titleAsHstring = titleAsStdWstring;`
- `std::wcout << titleAsHstring.c_str() << std::endl;`
- `}`
- `}`
- `#include <winrt/Windows.Foundation.Collections.h>`
- `#include <winrt/Windows.Web.Syndication.h>`Windows SDK, because your project (via the `cppwinrt.exe` tool) generates those same
- headers into your project's *$(GeneratedFilesDir)* folder. They'll be loaded from that
- folder if they can't be found elsewhere, or if you change your project settings.
- The headers contain Windows APIs projected into C++/WinRT. In other words, for each

- Windows type, C++/WinRT defines a C++-friendly equivalent (called the *projected type*).
- A projected type has the same fully-qualified name as the Windows type, but it's placed
- in the C++ **winrt** namespace. Putting these includes in your precompiled header
- reduces incremental build times.
- **Important**
- Whenever you want to use a type from a Windows namespaces, you must `#include`
- the corresponding C++/WinRT Windows namespace header file, as shown above.
- The *corresponding* header is the one with the same name as the type's namespace.
- For example, to use the C++/WinRT projection for the
- **Windows::Foundation::Collections::PropertySet** runtime class, include the
- `winrt/Windows.Foundation.Collections.h` header.
- It is common for a C++/WinRT projection header to automatically include related
- namespace header files. For example, `winrt/Windows.Foundation.Collections.h`
- includes `winrt/Windows.Foundation.h` . But you shouldn't rely on this behavior, since
- it's an implementation detail that changes over time. You must explicitly include
- any headers that you need.
- C++/WinRT
- `using namespace winrt;`
- `using namespace Windows::Foundation;`
- `using namespace Windows::Web::Syndication;`
- The `using namespace` directives are optional, but convenient. The pattern shown above
- for such directives (allowing unqualified name lookup for anything in the **winrt**
- namespace) is suitable for when you're beginning a new project and C++/WinRT is the
- only language projection you're using inside of that project. If, on the other hand, you're
- mixing C++/WinRT code with C++/CX and/or SDK application binary interface (ABI)
- code (you're either porting from, or interoperating with, one or both of those models),
- then see the topics Interop between C++/WinRT and C++/CX, Move to C++/WinRT
- from C++/CX, and Interop between C++/WinRT and the ABI.
- C++/WinRT`winrt::init_apartment();`

- The call to **winrt::init_apartment** initializes the thread in the Windows Runtime; by
- default, in a multithreaded apartment. The call also initializes COM.
- C++/WinRT
- `Uri rssFeedUri{ L"https://blogs.windows.com/feed" };`
- `SyndicationClient syndicationClient;`
- Stack-allocate two objects: they represent the uri of the Windows blog, and a
- syndication client. We construct the uri with a simple wide string literal (see String
- handling in C++/WinRT for more ways you can work with strings).
- C++/WinRT
- `SyndicationFeed syndicationFeed =`
- `syndicationClient.RetrieveFeedAsync(rssFeedUri).get();`
- **SyndicationClient::RetrieveFeedAsync** is an example of an asynchronous Windows
- Runtime function. The code example receives an asynchronous operation object from
- **RetrieveFeedAsync**, and it calls **get** on that object to block the calling thread and wait
- for the result (which is a syndication feed, in this case). For more about concurrency, and
- for non-blocking techniques, see Concurrency and asynchronous operations with
- C++/WinRT.
- C++/WinRT
- `for (const SyndicationItem syndicationItem : syndicationFeed.Items()) { ...`
- `}`
- **SyndicationFeed.Items** is a range, defined by the iterators returned from **begin** and **end**
- functions (or their constant, reverse, and constant-reverse variants). Because of this, you
- can enumerate **Items** with either a range-based `for` statement, or with the **std::for_each**
- template function. Whenever you iterate over a Windows Runtime collection like this,
- you'll need to `#include <winrt/Windows.Foundation.Collections.h>` .
- C++/WinRT
- `winrt::hstring titleAsHstring = syndicationItem.Title().Text();`
- `// Omitted: there's a little bit of extra work here to remove the trademark`
- `symbol from the title text.std::wcout << titleAsHstring.c_str() << std::endl;`
- Gets the feed's title text, as a **winrt::hstring** object (more details in String handling in
- C++/WinRT). The **hstring** is then output, via the **c_str** function, which reflects the pattern

- used with C++ Standard Library strings.
- As you can see, C++/WinRT encourages modern, and class-like, C++ expressions such
- as `syndicationItem.Title().Text()` . This is a different, and cleaner, programming style
- from traditional COM programming. You don't need to directly initialize COM, nor work
- with COM pointers.
- Nor do you need to handle HRESULT return codes. C++/WinRT converts error HRESULTs
- to exceptions such as **winrt::hresult-error** for a natural and modern programming style.
- For more info about error-handling, and code examples, see Error handling with
- C++/WinRT.

Modify a Windows Desktop application project to add C++/WinRT support

- Some desktop projects (for example, the WinUI 3 templates in Visual Studio) have
- C++/WinRT support built in.
- But this section shows you how you can add C++/WinRT support to any Windows
- Desktop application project that you might have. If you don't have an existing Windows
- Desktop application project, then you can follow along with these steps by first creating
- one. For example, open Visual Studio and create a **Visual C++** > **Windows Desktop** >
- **Windows Desktop Application** project.
- You can optionally install the C++/WinRT Visual Studio Extension (VSIX) and the
- NuGet package. For details, see Visual Studio support for C++/WinRT.

Set project properties

- Go to project property **General** > **Windows SDK Version**, and select **All Configurations**
- and **All Platforms**. Ensure that **Windows SDK Version** is set to 10.0.17134.0 (Windows
- 10, version 1803) or greater.
- Confirm that you're not affected by Why won't my new project compile?.
- Because C++/WinRT uses features from the C++17 standard, set project property

- **C/C++** > **Language** > **C++ Language Standard** to *ISO C++17 Standard (/std:c++17)*.
The precompiled header
- The default project template creates a precompiled header for you, named either
- `framework.h` , or `stdafx.h` . Rename that to `pch.h` . If you have a `stdafx.cpp` file, then
- rename that to `pch.cpp` . Set project property **C/C++** > **Precompiled Headers** >
- **Precompiled Header** to *Create (/Yc)*, and **Precompiled Header File** to *pch.h*.
- Find and replace all `#include "framework.h"` (or `#include "stdafx.h"`) with `#include`
- `"pch.h"` .
- In `pch.h` , include `winrt/base.h` .
- C++/WinRT
- `// pch.h`
- `...`
- `#include <winrt/base.h>`
Linking
- The C++/WinRT language projection depends on certain Windows Runtime free (non
- member) functions, and entry points, that require linking to the WindowsApp.lib
- umbrella library. This section describes three ways of satisfying the linker.
- The first option is to add to your Visual Studio project all of the C++/WinRT MSBuild
- properties and targets. To do this, install the Microsoft.Windows.CppWinRT NuGet
- package into your project. Open the project in Visual Studio, click **Project** > **Manage**
- **NuGet Packages...** > **Browse**, type or paste **Microsoft.Windows.CppWinRT** in the search
- box, select the item in search results, and then click **Install** to install the package for that
- project.
- You can also use project link settings to explicitly link `WindowsApp.lib` . Or, you can do it
- in source code (in `pch.h` , for example) like this.
- C++/WinRT
- `#pragma comment(lib, "windowsapp")`
- You can now compile and link, and add C++/WinRT code to your project (for example,
- code similar to that shown in the A C++/WinRT quick-start section, above).

- # The three main scenarios for C++/WinRT As you use and become familiar with C++/WinRT, and work through the rest of the
- documentation here, you'll likely notice that there are three main scenarios, as described
- in the following sections.
- # Consuming Windows APIs and types
- In other words, *using*, or *calling* APIs. For example, making API calls to communicate
- using Bluetooth; to stream and present video; to integrate with the Windows shell; and
- so on. C++/WinRT fully and uncompromisingly supports this category of scenario. For
- more info, see Consume APIs with C++/WinRT.
- # Authoring Windows APIs and types
- In other words, *producing* APIs and types. For example, producing the kinds of APIs
- described in the section above; or the graphics APIs; the storage and file system APIs;
- the networking APIs, and so on. For more info, see Author APIs with C++/WinRT.
- Authoring APIs with C++/WinRT is a little more involved than consuming them, because
- you must use IDL to define the shape of the API before you can implement it. There's a
- walkthrough of doing that in XAML controls; bind to a C++/WinRT property.
- # XAML applications
- This scenario is about building applications and controls on the XAML UI framework.
- Working in a XAML application amounts to a combination of consuming and authoring.
- But since XAML is the dominant UI framework on Windows today, and its influence over
- the Windows Runtime is proportionate to that, it deserves its own category of scenario.
- Be aware that XAML works best with programming languages that offer reflection. In
- C++/WinRT, you sometimes have to do a little extra work in order to interoperate with
- the XAML framework. All of those cases are covered in the documentation. Good places

- to start are XAML controls; bind to a C++/WinRT property and XAML custom
- (templated) controls with C++/WinRT.

Sample apps written in C++/WinRT

- See Where can I find C++/WinRT sample apps?.

Important APIsFeedback

- **Was this page helpful?**
- Provide product feedback
- | Get help at Microsoft Q&A
- SyndicationClient::RetrieveFeedAsync method
- SyndicationFeed.Items property
- winrt::hstring struct
- winrt::hresult-error struct
- C++/CX
- Error handling with C++/WinRT
- Interop between C++/WinRT and C++/CX
- Interop between C++/WinRT and the ABI
- Move to C++/WinRT from C++/CX
- String handling in C++/WinRT

Related topics

- **Yes**

NoFeedback

- **Was this page helpful?**
- Get help at Microsoft Q&A

Get Started with Win32 and C++

- Article • 01/27/2022
- The aim of this Get Started series is to teach you how to write a desktop program in
- C++ using Win32 and COM APIs.
- In the first module, you'll learn step-by-step how to create and show a window. Later
- modules will introduce the Component Object Model (COM), graphics and text, and
- user input.
- For this series, it is assumed that you have a good working knowledge of C++
- programming. No previous experience with Windows programming is assumed. If you
- are new to C++, learning material is available in the C++ language documentation .

- **Topic**
- **Description**
- Intro to Win32
- programming in C++
- This section describes some of the basic terminology and coding
- conventions used in Windows programming.
- Module 1. Your First
- Windows Program
- In this module, you will create a simple Windows program that
- shows a blank window.
- Module 2. Using COM in
- Your Windows Program
- This module introduces the Component Object Model (COM),
- which underlies many of the modern Windows APIs.
- Module 3. Windows
- Graphics
- This module introduces the Windows graphics architecture, with a
- focus on Direct2D.
- Module 4. User Input
- This module describes mouse and keyboard input.
- Sample Code
- Contains links to download the sample code for this series.
- # In this section
- **Yes**
- **No**# Create a simple Universal Windows
- # Platform (UWP) game with DirectX
- Article • 10/20/2022
- In this set of tutorials, you'll learn how to use DirectX and C++/WinRT to create the basic
- Universal Windows Platform (UWP) sample game named **Simple3DGameDX**. The
- gameplay takes place in a simple first-person 3D shooting gallery.
- **Note**
- The link from which you can download the **Simple3DGameDX** sample game itself is
- **Direct3D sample game**. The C++/WinRT source code is in the folder named
- `cppwinrt` . For info about other UWP sample apps, see **Sample applications for**
- **Windows development**.
- These tutorials cover all of the major parts of a game, including the processes for

- loading assets such as arts and meshes, creating a main game loop, implementing a
- simple rendering pipeline, and adding sound and controls.
- You'll also see UWP game development techniques and considerations. We'll focus on
- key UWP DirectX game development concepts, and call out Windows-Runtime-specific
- considerations around those concepts.

Objective

- To learn about the basic concepts and components of a UWP DirectX game, and to
- become more comfortable designing UWP games with DirectX.

What you need to know

- For this tutorial, you need to be familiar with these subjects.
- C++/WinRT. C++/WinRT is a standard modern C++17 language projection for
- Windows APIs, implemented as a header-file-based library, and designed to
- provide you with first-class access to the modern Windows APIs.
- Basic linear algebra and Newtonian physics concepts.
- Basic graphics programming terminology.
- Basic Windows programming concepts.
- Basic familiarity with the Direct2D and Direct3D 11 APIs.The **Simple3DGameDX** sample game implements a simple first-person 3D shooting
- gallery, where the player fires balls at moving targets. Hitting each target awards a set
- number of points, and the player can progress through 6 levels of increasing challenge.
- At the end of the levels, the points are tallied, and the player is awarded a final score.
- The sample demonstrates these game concepts.
- Interoperation between DirectX 11.1 and the Windows Runtime
- A first-person 3D perspective and camera
- Stereoscopic 3D effects
- Collision-detection between objects in 3D
- Handling player input for mouse, touch, and Xbox controller controls
- Audio mixing and playback
- A basic game state-machine
- **Topic**
- **Description**
- Set up the
- game project
- The first step in developing your game is to set up a project in Microsoft Visual
- Studio. After you've configured a project specifically for game development, you

- could later re-use it as a kind of template.
- Define the
- game's UWP
- app framework
- The first step in coding a Universal Windows Platform (UWP) game is building
- the framework that lets the app object interact with Windows.
- Game flow
- management
- Define the high-level state machine to enable player and system interaction.
- Learn how UI interacts with the overall game's state machine and how to create
- event handlers for UWP games.
- # Direct3D UWP shooting gallery sample Topic
- **Description**
- Define the
- main game
- object
- Now, we look at the details of the sample game's main object and how the rules
- it implements translate into interactions with the game world.
- Rendering
- framework I:
- Intro to
- rendering
- Learn how to develop the rendering pipeline to display graphics. Intro to
- rendering.
- Rendering
- framework II:
- Game
- rendering
- Learn how to assemble the rendering pipeline to display graphics. Game
- rendering, set up and prepare data.
- Add a user
- interface
- Learn how to add a 2D user interface overlay to a DirectX UWP game.
- Add controls
- Now, we take a look at how the sample game implements move-look controls
- in a 3-D game, and how to develop basic touch, mouse, and game controller
- controls.
- Add sound
- Develop a simple sound engine using XAudio2 APIs to playback game music
- and sound effects.
- Extend the
- sample game
- Learn how to implement a XAML overlay for a UWP DirectX
  game.# Create a console calculator in C++
- Article • 10/08/2024

- The usual starting point for a C++ programmer is a "Hello, world!" application that runs
- on the command line. You start with that in this article, and then move on to something
- more challenging: a calculator app.

Prerequisites

- Visual Studio with the **Desktop development with C++** workload installed and
- running on your computer. To install it, see Install C++ support in Visual Studio.
- This tutorial demonstrates a feature called edit and continue which allows you to
- make changes to your code while the app is running. To enable edit and continue,
- from the main menu select **Tools** > **Options** > **Debugging** > **General** and ensure
- that **Require source files to exactly match the original version** is checked.

Create your app project

- Visual Studio uses *projects* to organize the code for an app, and *solutions* to organize
- one or more projects. A project contains all the options, configurations, and rules used
- to build an app. It also manages the relationship between all the project's files and any
- external files. To create your app, first, create a new project and solution.
- 1. Start Visual Studio--the Visual Studio Start dialog box appears. Select **Create a new**
- **project** to get started.2. In the **Create a new project** dialog, set the language dropdown to **C++**, set the
- platform dropdown to **Windows**, select **Console App** from the list of project types,
- then select **Next**.
- **Important**
- Make sure you select the C++ version of the **Console App** template. It has the
- **C++**, **Windows**, and **Console** tags, and the icon has "++" in the corner.3. In the **Configure your new project** dialog box, select the **Project name** text box,
- name your new project *CalculatorTutorial*, then select **Create**.
- An empty C++ Windows console application 'Hello World' app is created. Console

- applications use a Windows console window to display output and accept user
- input. In Visual Studio, an editor window opens and shows the generated code:
- C++
- // CalculatorTutorial.cpp : This file contains the 'main' function.
- Program execution begins and ends there.
- //
- #include <iostream>
- int main()
- {
- std::cout << "Hello World!\n";
- }
- // Run program: Ctrl + F5 or Debug > Start Without Debugging menu
- // Debug program: F5 or Debug > Start Debugging menu
- // Tips for Getting Started:
- // 1. Use the Solution Explorer window to add/manage files
- // 2. Use the Team Explorer window to connect to source control
- // 3. Use the Output window to see build output and other messages
- // 4. Use the Error List window to view errors
- // 5. Go to Project > Add New Item to create new code files, or
- Project > Add Existing Item to add existing code files to the project// 6. In the future, to open this project again, go to File > Open >
- Project and select the .sln file
- # Verify that your new app builds and runs
- The template for a new Windows console application creates a simple C++ "Hello
- World" app. At this point, you can see how Visual Studio builds and runs the apps you
- create right from the IDE.
- 1. To build your project, select **Build Solution** from the **Build** menu. The **Output**
- window shows the results of the build process.
- 2. To run the code, on the menu bar, select **Debug** > **Start without debugging**
- (Ctrl+F5).A console window opens and your app runs within it.
- When you start a console app in Visual Studio, it runs your code, then prints "Press
- any key to close this window . . ." to give you a chance to see the output.
- Congratulations! You created your first "Hello, world!" console app in Visual Studio!
- 3. Press a key to dismiss the console window and return to Visual Studio.

- You now have the tools to build and run your app after every change, to verify that the
- code still works as you expect. Later, we show you how to debug it if it doesn't.
- Now let's modify the code in this template to be a calculator app.
- 1. Replace the contents of the *CalculatorTutorial.cpp* file with the following code so
- that it matches this example:
- C++
- # Edit the code
- `// CalculatorTutorial.cpp : This file contains the 'main' function.`
- `Program execution begins and ends there.`
- `//`
- `#include <iostream>`
- `using namespace std;`
- `int main()`
- `{`
- `cout << "Calculator Console Application" << endl << endl;`
- `cout << "Please enter the operation to perform. Format: a+b | a-b |`
  Understanding the code:
- The `#include` statement brings in code in other files. Sometimes, you
- may see a filename surrounded by angle brackets like `<iostream>` . The
- angle brackets instruct the compiler to look for the `iostream` header file
- first in the standard system directories, and if not found, to look in
- directories specific to the project. Other times, you may see a filename
- surrounded by quotes like `"someHeader.h"` . The quotes instruct the
- compiler to skip looking in the standard system directories and instead
- only look in directories specific to the project.
- The `using namespace std;` tells the compiler to expect code from the
- C++ Standard Library to be used in this file. Without this line, each
- keyword from the library would have to be preceded with `std::` to
- denote its scope. For instance, without that line, each reference to `cout`
- would be written as `std::cout` . The **using** statement is added to make it
- more convenient to access code in another namespace.
- The `cout` keyword is used to print to standard output in C++. The `<<`
- operator tells the compiler to send whatever is to the right of it to the
- standard output.
- The `endl` keyword is like the Enter key; it ends the line and moves the
- cursor to the next line. It's a better practice to put a `\n` inside the string
- (contained by `""`) to do the same thing because `endl` always flushes the
- buffer which can hurt the performance of the program. But since this is a
- very small app, `endl` is used instead.

- All C++ statements must end with semicolons and all C++ applications
- must contain a `main()` function. This function is what the program runs at
- `a*b | a/b"`
- `<< endl;`
- `return 0;`
- `}`
- `// Run program: Ctrl + F5 or Debug > Start Without Debugging menu`
- `// Debug program: F5 or Debug > Start Debugging menu`
- `// Tips for Getting Started:`
- `// 1. Use the Solution Explorer window to add/manage files`
- `// 2. Use the Team Explorer window to connect to source control`
- `// 3. Use the Output window to see build output and other messages`
- `// 4. Use the Error List window to view errors`
- `// 5. Go to Project > Add New Item to create new code files, or`
- `Project > Add Existing Item to add existing code files to the project`
- `// 6. In the future, to open this project again, go to File > Open >`
- `Project and select the .sln file`the start. All code must be accessible from `main()` in order to be used.
- 2. To save the file, press **Ctrl+S**, or select the floppy disk icon in the toolbar under the
- menu bar.
- 3. To run the application, press **Ctrl+F5** or go to the **Debug** menu and select **Start**
- **Without Debugging**. You should see a console window appear that looks like this.
- 4. Close the console window when you're done.

Add code to do some math

- A class is like a blueprint for an object that does something. In this case, we define a
- calculator class to contain the math logic.

Add a Calculator class

- 1. Go to the **Project** menu and select **Add Class**. In the **Class Name** edit box, enter
- *Calculator*. Select **OK**.
- Two new files get added to your project. To save all your changed files at once,
- press **Ctrl+Shift+S**. It's a keyboard shortcut for **File** > **Save All**. There's also a
- toolbar button for **Save All**, an icon of two floppy disks, found beside the **Save**button. In general, it's good practice to do **Save All** frequently, so you don't miss
- saving any changes.

- The **Add Class** wizard creates `.h` and `.cpp` files that have the same name as the
- class. You can see a full list of your project files in the **Solution Explorer** window,
- visible on the side of the IDE. If the window isn't visible, open it from the menu bar
- via **View** > **Solution Explorer**.
- You can open a file by double-clicking it in the **Solution Explorer** window. Double
- click `Calculator.h` to open it.
- 2. Replace the contents of **Calculator.h** with the following code so that the file now
- looks like this:
- C++
- Understanding the code
- This code declares a new function called `Calculate` , which handles math
- operations for addition, subtraction, multiplication, and division.
- `#pragma once`
- `class Calculator`
- `{`
- `public:`
- `double Calculate(double x, char oper, double y);`
- `};`C++ code is organized into *header* (`.h`) files and *source* (`.cpp`) files.
- Some other file extensions are supported by various compilers, but these
- are the main ones to know about. Functions and variables are normally
- *declared*, that is, given a name and a type, in header files, and
- *implemented*, or given a definition, in source files. To access code defined
- in another file, you can use `#include "filename.h"` , where `filename.h` is
- the name of the file that declares the variables or functions you want to
- use.
- It's good practice to organize your code into different files based on what
- it does, so it's easy to find the code you need later. In our case, we define
- the `Calculator` class separately from the file containing the `main()`
- function, but we plan to reference the `Calculator` class in `main()` .
- 3. A green squiggle appears under `Calculate` because although the `Calculate`
- function is *declared*, it isn't *defined*. Hover over `Calculate` , click the down arrow on
- the screwdriver icon, and select **Create definition of 'Calculate' in Calculator.cpp** .

- This code is added to *Calculator.cpp* :Currently, it just returns 0.0. Let's change that.
- 4. Switch to the *Calculator.cpp* file in the editor window. Replace the contents of
- *Calculator::Calculate(double x, char oper, double y)* with:
- C++
- Understanding the code
- The function `Calculate` takes a number, an operator, and a second
- number. Then it performs the requested operation on the two numbers.
- The `switch` statement checks which operator was provided, and executes
- the case corresponding to that operation. The `default:` case is a fallback
- `double Calculator::Calculate(double x, char oper, double y)`
- `{`
- `switch(oper)`
- `{`
- `case '+':`
- `return x + y;`
- `case '-':`
- `return x - y;`
- `case '*':`
- `return x * y;`
- `case '/':`
- `return x / y;`
- `default:`
- `return 0.0;`
- `}`
- `}`in case the user types an operator that isn't handled by any of the
- preceding `case` statements. It's best to handle invalid user input in a
- more elegant way, but this is beyond the scope of this tutorial.
- The **double** keyword denotes a type of number that supports decimals.
- This type of number is called a floating-point number, and `double` means
- a floating point number that has extra precision. This way, the calculator
- can handle both decimal math and integer math. The `Calculate` function
- is required to always return a double-precision floating point number due
- to the **double** at the start of the code (this denotes the function's return
- type), which is why we return 0.0 in the default case.
- The `.h` file declares the function *prototype*, which tells the compiler
- upfront what parameters it requires, and what return type to expect from
- it. The `.cpp` file has all the implementation details of the function.

- If you build and run the code again at this point, it immediately exits after asking which
- operation to perform. So, modify the `main` function to do multiple calculations.
- 1. Update the `main` function in *CalculatorTutorial.cpp* as follows:
- C++
- # Call the Calculator class member functions
- // CalculatorTutorial.cpp : This file contains the 'main' function.
- Program execution begins and ends there.
- //
- #include <iostream>
- #include "Calculator.h"
- using namespace std;
- int main()
- {
- double x = 0.0;
- double y = 0.0;
- double result = 0.0;
- char oper = '+';
- cout << "Calculator Console Application" << endl << endl;
- cout << "Please enter the operation to perform. Format: a+b | a-b |
- a*b | a/b"
- << endl;
- Calculator c;
- while (true)
- {Understanding the code
- Since C++ programs always start at the `main()` function, we need to call
- our other code from there, so an `#include` statement is needed to make
- that code visible to our `main()` function.
- The variables `x` , `y` , `oper` , and `result` are declared to store the first
- number, second number, operator, and final result, respectively. It's
- always good practice to give them some initial values to avoid undefined
- behavior, which is what is done here.
- The `Calculator c;` line declares an object named `c` as an instance of the
- `Calculator` class. The class itself is just a blueprint for how calculators
- work; the object is the specific calculator that does the math.
- The `while (true)` statement is a loop. The code inside the loop executes
- over and over again as long as the condition inside the `()` holds true.
- Since the condition is simply listed as **true** , it's always true, so the loop
- runs forever. To close the program, the user must manually close the
- console window. Otherwise, the program always waits for new input.
- The `cin` keyword accepts input from the user. The input stream is smart
- enough to process a line of text entered in the console window and place
- it inside each of the variables listed, in order.

- The `c.Calculate(x, oper, y);` expression calls the `Calculate` function
- defined earlier, and supplies the entered input values and the requested
- operation. The function then returns a number that is stored in `result` .
- Finally, `result` is printed to the console and the user sees the result of
- the calculation.
- Now test the program again to make sure everything works properly.
- 1. Press **Ctrl+F5** to rebuild and start the app.
- 2. Enter `5+5` , and press **Enter**. Verify that the result is 10.
- `cin >> x >> oper >> y;`
- `result = c.Calculate(x, oper, y);`
- `cout << "Result " << "of " << x << oper << y << " is: " <<`
- `result << endl;`
- `}`
- `return 0;`
- `}`
- **Build and test the code again**3. Stop the program by closing the console window.
- # Debug the app
- Since the user is free to type anything into the console window, let's make sure the
- calculator handles unexpected input. Instead of running the program, let's debug it so
- we can inspect what it's doing step-by-step.
- # Run the app in the debugger
- 1. In `CalcuatorTutorial.cpp` , set a breakpoint on the line: `result = c.Calculate(x,`
- `oper, y);` . To set the breakpoint, click next to the line in the gray vertical bar along
- the left edge of the editor window so that a red dot appears.
- Now when we debug the program, execution pauses at that line. We already have
- a rough idea that the program works for simple cases. Since we don't want to
- pause execution every time we call `Calculate()` , let's make the breakpoint
- conditional.2. Right-click the red dot that represents the breakpoint, and select **Conditions**. In
- the edit box for the condition, enter `(y == 0) && (oper == '/')` . Select the **Close**
- button to save the breakpoint condition.
- Now, execution pauses at the breakpoint when the app tries to divide by 0.
- 3. To debug the program, press **F5**, or select the **Local Windows Debugger** debugger

- toolbar button that has the green arrow icon. In your console app, if you enter
- something like "5 - 0", the program behaves normally and keeps running.
- However, if you type "10 / 0", it pauses at the breakpoint. You can put any number
- of spaces between the operator and numbers: `cin` is smart enough to parse the
- input appropriately.

Useful windows in the debugger

- When you debug your code, you may notice that some new windows appear. These
- windows can assist your debugging experience. Take a look at the **Autos** window. The
- **Autos** window shows you the current values of variables used at least three lines before
- and up to the current line. If you don't see the **Autos** window, from the main menu
- select **Debug** > **Windows** > **Autos**.
- To see all of the variables from that function, switch to the **Locals** window. Because this
- is a small function, the Autos and Locals window show the same variables. But you can
- modify the values of these variables in the Locals window while debugging to see what
- effect they would have on the program. In this case, we leave them alone. Open the
- **Locals** window by selecting **Locals** at the bottom of the **Autos** window, or by selecting
- from the main menu **Debug** > **Windows** > **Locals**.You can also hover over variables in the code to see their current values at the point
- where execution is currently paused. Make sure the editor window is in focus by clicking
- on it first.

Continue debugging

- 1. The yellow arrow on the left shows the current point of execution. The current line
- calls `Calculate` , so press **F11** to **Step Into** the function. Now you're executing code
- in the body of the `Calculate` function. Be careful with **Step Into** because it steps
- into any functions on the line you're on, including standard library functions. It's

- fine to step into the standard library, but you may be more interested in focusing
- on your code instead of library code.
- 2. Now that the point of execution is at the start of the `Calculate` function, press **F10**
- to move to the next line in the program's execution. **F10** is also known as **Step**
- **Over**. You can use **Step Over** to move from line to line, without delving into the
- details of what is occurring in each part of the line. In general, you should use **Step**
- **Over** instead of **Step Into** unless you want to dive more deeply into code that is
- being called from elsewhere (as you did to reach the body of `Calculate`).
- 3. Continue using **F10** to **Step Over** each line until you get back to the `main()`
- function in the other file, and stop on the `cout` line.The program is doing what's expected: it takes the first number, and divides it by
- the second. On the `cout` line, hover over the `result` variable or take a look at
- `result` in the **Autos** window. Its value is `inf` , which doesn't look right.
- Let's fix it. The `cout` line outputs whatever value is stored in `result` , so when you
- step one more line forward using **F10**, the console window displays:
- This result is because division by zero is undefined, so the program doesn't have a
- numerical answer for the requested operation.
- # Fix the "divide by zero" error
- Let's handle division by zero more gracefully so that it's easier for the user to
- understand the problem.
- 1. Make the following changes in *CalculatorTutorial.cpp* . You can leave the
- program running as you edit, thanks to a debugger feature called **Edit andContinue**. Add an `if` statement following `cin >> x >> oper >> y;` to check for
- division by zero and output a message to the user if it happens. Otherwise, the
- result is printed.
- C++
- 2. Press **F5** once. Program execution continues until it has to pause to ask for user
- input. Enter `10 / 0` again. Now, a more helpful message is printed. The user is
- asked for more input, and the program continues executing normally.
- `// CalculatorTutorial.cpp : This file contains the 'main' function.`

```
Program execution begins and ends there.
//
#include <iostream>
#include "Calculator.h"
using namespace std;
int main()
{
double x = 0.0;
double y = 0.0;
double result = 0.0;
char oper = '+';
cout << "Calculator Console Application" << endl << endl;
cout << "Please enter the operation to perform. Format: a+b | a-b |
a*b | a/b" << endl;
Calculator c;
while (true)
{
cin >> x >> oper >> y;
if (oper == '/' && y == 0)
{
cout << "Math error: Attempted to divide by zero!" << endl;
continue;
}
else
{
result = c.Calculate(x, oper, y);
}
cout << "Result " << "of " << x << oper << y << " is: " <<
result << endl;
}
return 0;
}
```

Note

When you edit code while in debugging mode, there's a risk of code becoming stale. This happens when the debugger is still running your old code, and has not yet updated it with your changes. The debugger displays a dialog to inform you when this happens. Sometimes, you may need to press **F5** to refresh the code being executed. In particular, if you make a change inside a function while the point of execution is inside that function, you need to step out of the function, then back into it again to get the updated code. If that doesn't work and you see an error message, you can stop debugging by clicking on the red square in the toolbar under the menus at the top of the

- IDE, then start debugging again by entering **F5** or by choosing the green
- "play" arrow beside the stop button on the toolbar.
- Another reason edit and continue may fail is if you see a message that says
- "The Require source files to exactly match the original version setting under
- Debug->Options->General needs to be enabled..." To fix this, from the main
- menu select **Tools** > **Options** > **Debugging** > **General** and ensure that
- **Require source files to exactly match the original version** is checked.
- Understanding the Run and Debug shortcuts
- **F5**, or **Debug** > **Start Debugging**, starts a debugging session, if one isn't
- already active, and runs the program until a breakpoint is hit or the
- program needs user input. If no user input is needed and no breakpoint
- is available to hit, the program terminates and the console window closes
- itself when the program finishes running. If your program outputs to the
- console, use **Ctrl+F5** or set a breakpoint before you press **F5** to keep the
- window open.
- **Ctrl+F5**, or **Debug** > **Start Without Debugging**, runs the application
- without going into debug mode. This is slightly faster than debugging,
- and the console window stays open after the program finishes executing.
- **F10**, known as **Step Over**, lets you iterate through code, line-by-line, and
- visualize how the code is run and what variable values are at each step of
- execution.**F11**, known as **Step Into**, works similarly to **Step Over**, except it steps into
- any functions called on the line of execution. For example, if the line
- being executed calls a function, pressing **F11** moves the pointer into the
- body of the function, so you can follow the function's code being run
- before coming back to the line you started at. Pressing **F10** steps over the
- function call and just moves to the next line; the function call still
- happens, but the program doesn't pause to show you what it's doing.
- # Close the app

- If it's still running, close the console window to stop the calculator app.

Add Git source control

- Now that you've created an app, you might want to add it to a Git repository. We've got
- you covered. Visual Studio makes that process easy with Git tools you can use directly
- from the IDE.
- **Tip**
- Git is the most widely used modern version control system, so whether you're a
- professional developer or you're learning how to code, Git can be very useful. If
- you're new to Git, the **https://git-scm.com/** website is a good place to start.
- There, you can find cheat sheets, a popular online book, and Git Basics videos.
- To associate your code with Git, you start by creating a new Git repository where your
- code is located. Here's how:
- 1. In the status bar at the bottom-right corner of Visual Studio, select **Add to Source**
- **Control**, and then select **Git**.
- 2. In the **Create a Git repository** dialog box, sign in to GitHub.The repository name auto-populates based on your folder location. By default,
- your new repository is private, which means you're the only one who can access it.
- **Tip**
- Whether your repository is public or private, it's best to have a remote backup
- of your code stored securely on GitHub. Even if you aren't working with a
- team, a remote repository makes your code available to you from any
- computer.
- 3. Select **Create and Push**.
- After you create your repository, status details appear in the status bar.
- The first icon with the arrows shows how many outgoing/incoming commits are in
- your current branch. You can use this icon to pull any incoming commits or push
- any outgoing commits. You can also choose to view these commits first. To do so,

- select the icon, and then select **View Outgoing/Incoming**.

Feedback

- **Was this page helpful?**
- Provide product feedback
- | Get help at Microsoft Q&A
- The second icon with the pencil shows the number of uncommitted changes to
- your code. You can select this icon to view those changes in the **Git Changes**
- window.
- To learn more about how to use Git with your app, see the Visual Studio version control
- documentation.
- Congratulations! You completed the code for the calculator app, built and debugged it,
- and added it to a repo, all in Visual Studio.
- Learn more about Visual Studio for C++

The finished app

Next steps

- **Yes**
- **No**

Get started with C++/WinRT

- Article • 02/13/2023
- **Important**
- For info about setting up Visual Studio for C++/WinRT development—including
- installing and using the C++/WinRT Visual Studio Extension (VSIX) and the NuGet
- package (which together provide project template and build support)—see **Visual**
- **Studio support for C++/WinRT**.
- To get you up to speed with using C++/WinRT, this topic walks through a simple code
- example based on a new **Windows Console Application (C++/WinRT)** project. This
- topic also shows how to add C++/WinRT support to a Windows Desktop application
- project.
- **Note**
- While we recommend that you develop with the latest versions of Visual Studio and
- the Windows SDK, if you're using Visual Studio 2017 (version 15.8.0 or later), and

- targeting the Windows SDK version 10.0.17134.0 (Windows 10, version 1803), then
- a newly created C++/WinRT project may fail to compile with the error "*error C3861:*
- *'from_abi': identifier not found*", and with other errors originating in *base.h*. The
- solution is to either target a later (more conformant) version of the Windows SDK,
- or set project property **C/C++** > **Language** > **Conformance mode: No** (also, if
- **/permissive-** appears in project property **C/C++** > **Language** > **Command Line**
- under **Additional Options**, then delete it).
- # A C++/WinRT quick-start
- Create a new **Windows Console Application (C++/WinRT)** project.
- Edit `pch.h` and `main.cpp` to look like this.
- C++/WinRT
- `// pch.h`
- `#pragma once`
- `#include <winrt/Windows.Foundation.Collections.h>`C++/WinRT
- Let's take the short code example above piece by piece, and explain what's going on in
- each part.
- C++/WinRT
- With the default project settings, the included headers come from the Windows SDK,
- inside the folder
- `%WindowsSdkDir%Include<WindowsTargetPlatformVersion>\cppwinrt\winrt .` Visual Studio
- includes that path in its *IncludePath* macro. But there's no strict dependency on the
- `#include <winrt/Windows.Web.Syndication.h>`
- `#include <iostream>`
- `// main.cpp`
- `#include "pch.h"`
- `using namespace winrt;`
- `using namespace Windows::Foundation;`
- `using namespace Windows::Web::Syndication;`
- `int main()`
- `{`
- `winrt::init_apartment();`
- `Uri rssFeedUri{ L"https://blogs.windows.com/feed" };`
- `SyndicationClient syndicationClient;`
- `syndicationClient.SetRequestHeader(L"User-Agent", L"Mozilla/5.0`
- `(compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)");`
- `SyndicationFeed syndicationFeed =`
- `syndicationClient.RetrieveFeedAsync(rssFeedUri).get();`

- `for (const SyndicationItem syndicationItem : syndicationFeed.Items())`
- {
- `winrt::hstring titleAsHstring = syndicationItem.Title().Text();`
- `// A workaround to remove the trademark symbol from the title`
- `string, because it causes issues in this case.`
- `std::wstring titleAsStdWstring{ titleAsHstring.c_str() };`
- `titleAsStdWstring.erase(remove(titleAsStdWstring.begin(),`
- `titleAsStdWstring.end(), L'™'), titleAsStdWstring.end());`
- `titleAsHstring = titleAsStdWstring;`
- `std::wcout << titleAsHstring.c_str() << std::endl;`
- }
- }
- `#include <winrt/Windows.Foundation.Collections.h>`
- `#include <winrt/Windows.Web.Syndication.h>`Windows SDK, because your project (via the `cppwinrt.exe` tool) generates those same
- headers into your project's *$(GeneratedFilesDir)* folder. They'll be loaded from that
- folder if they can't be found elsewhere, or if you change your project settings.
- The headers contain Windows APIs projected into C++/WinRT. In other words, for each
- Windows type, C++/WinRT defines a C++-friendly equivalent (called the *projected type*).
- A projected type has the same fully-qualified name as the Windows type, but it's placed
- in the C++ **winrt** namespace. Putting these includes in your precompiled header
- reduces incremental build times.
- **Important**
- Whenever you want to use a type from a Windows namespaces, you must `#include`
- the corresponding C++/WinRT Windows namespace header file, as shown above.
- The *corresponding* header is the one with the same name as the type's namespace.
- For example, to use the C++/WinRT projection for the
- **Windows::Foundation::Collections::PropertySet** runtime class, include the
- `winrt/Windows.Foundation.Collections.h` header.
- It is common for a C++/WinRT projection header to automatically include related
- namespace header files. For example, `winrt/Windows.Foundation.Collections.h`
- includes `winrt/Windows.Foundation.h` . But you shouldn't rely on this behavior, since
- it's an implementation detail that changes over time. You must explicitly include

- any headers that you need.
- C++/WinRT
- `using namespace winrt;`
- `using namespace Windows::Foundation;`
- `using namespace Windows::Web::Syndication;`
- The `using namespace` directives are optional, but convenient. The pattern shown above
- for such directives (allowing unqualified name lookup for anything in the **winrt**
- namespace) is suitable for when you're beginning a new project and C++/WinRT is the
- only language projection you're using inside of that project. If, on the other hand, you're
- mixing C++/WinRT code with C++/CX and/or SDK application binary interface (ABI)
- code (you're either porting from, or interoperating with, one or both of those models),
- then see the topics Interop between C++/WinRT and C++/CX, Move to C++/WinRT
- from C++/CX, and Interop between C++/WinRT and the ABI.
- C++/WinRT`winrt::init_apartment();`
- The call to **winrt::init_apartment** initializes the thread in the Windows Runtime; by
- default, in a multithreaded apartment. The call also initializes COM.
- C++/WinRT
- `Uri rssFeedUri{ L"https://blogs.windows.com/feed" };`
- `SyndicationClient syndicationClient;`
- Stack-allocate two objects: they represent the uri of the Windows blog, and a
- syndication client. We construct the uri with a simple wide string literal (see String
- handling in C++/WinRT for more ways you can work with strings).
- C++/WinRT
- `SyndicationFeed syndicationFeed =`
- `syndicationClient.RetrieveFeedAsync(rssFeedUri).get();`
- **SyndicationClient::RetrieveFeedAsync** is an example of an asynchronous Windows
- Runtime function. The code example receives an asynchronous operation object from
- **RetrieveFeedAsync**, and it calls **get** on that object to block the calling thread and wait
- for the result (which is a syndication feed, in this case). For more about concurrency, and
- for non-blocking techniques, see Concurrency and asynchronous operations with
- C++/WinRT.
- C++/WinRT

- `for (const SyndicationItem syndicationItem : syndicationFeed.Items()) { ...`
- `}`
- **SyndicationFeed.Items** is a range, defined by the iterators returned from **begin** and **end**
- functions (or their constant, reverse, and constant-reverse variants). Because of this, you
- can enumerate **Items** with either a range-based `for` statement, or with the **std::for_each**
- template function. Whenever you iterate over a Windows Runtime collection like this,
- you'll need to `#include <winrt/Windows.Foundation.Collections.h>` .
- C++/WinRT
- `winrt::hstring titleAsHstring = syndicationItem.Title().Text();`
- `// Omitted: there's a little bit of extra work here to remove the trademark`
- `symbol from the title text.std::wcout << titleAsHstring.c_str() << std::endl;`
- Gets the feed's title text, as a **winrt::hstring** object (more details in String handling in
- C++/WinRT). The **hstring** is then output, via the **c_str** function, which reflects the pattern
- used with C++ Standard Library strings.
- As you can see, C++/WinRT encourages modern, and class-like, C++ expressions such
- as `syndicationItem.Title().Text()` . This is a different, and cleaner, programming style
- from traditional COM programming. You don't need to directly initialize COM, nor work
- with COM pointers.
- Nor do you need to handle HRESULT return codes. C++/WinRT converts error HRESULTs
- to exceptions such as **winrt::hresult-error** for a natural and modern programming style.
- For more info about error-handling, and code examples, see Error handling with
- C++/WinRT.
- # Modify a Windows Desktop application project
- # to add C++/WinRT support
- Some desktop projects (for example, the WinUI 3 templates in Visual Studio) have
- C++/WinRT support built in.
- But this section shows you how you can add C++/WinRT support to any Windows

- Desktop application project that you might have. If you don't have an existing Windows
- Desktop application project, then you can follow along with these steps by first creating
- one. For example, open Visual Studio and create a **Visual C++** > **Windows Desktop** >
- **Windows Desktop Application** project.
- You can optionally install the C++/WinRT Visual Studio Extension (VSIX) and the
- NuGet package. For details, see Visual Studio support for C++/WinRT.

Set project properties

- Go to project property **General** > **Windows SDK Version**, and select **All Configurations**
- and **All Platforms**. Ensure that **Windows SDK Version** is set to 10.0.17134.0 (Windows
- 10, version 1803) or greater.
- Confirm that you're not affected by Why won't my new project compile?.
- Because C++/WinRT uses features from the C++17 standard, set project property
- **C/C++** > **Language** > **C++ Language Standard** to *ISO C++17 Standard (/std:c++17)*.

The precompiled header

- The default project template creates a precompiled header for you, named either
- `framework.h` , or `stdafx.h` . Rename that to `pch.h` . If you have a `stdafx.cpp` file, then
- rename that to `pch.cpp` . Set project property **C/C++** > **Precompiled Headers** >
- **Precompiled Header** to *Create (/Yc)*, and **Precompiled Header File** to *pch.h*.
- Find and replace all `#include "framework.h"` (or `#include "stdafx.h"`) with `#include`
- `"pch.h"` .
- In `pch.h` , include `winrt/base.h` .
- C++/WinRT
- `// pch.h`
- `...`
- `#include <winrt/base.h>`

Linking

- The C++/WinRT language projection depends on certain Windows Runtime free (non
- member) functions, and entry points, that require linking to the WindowsApp.lib
- umbrella library. This section describes three ways of satisfying the linker.

- The first option is to add to your Visual Studio project all of the C++/WinRT MSBuild
- properties and targets. To do this, install the Microsoft.Windows.CppWinRT NuGet
- package into your project. Open the project in Visual Studio, click **Project** > **Manage**
- **NuGet Packages...** > **Browse**, type or paste **Microsoft.Windows.CppWinRT** in the search
- box, select the item in search results, and then click **Install** to install the package for that
- project.
- You can also use project link settings to explicitly link `WindowsApp.lib` . Or, you can do it
- in source code (in `pch.h` , for example) like this.
- C++/WinRT
- `#pragma comment(lib, "windowsapp")`
- You can now compile and link, and add C++/WinRT code to your project (for example,
- code similar to that shown in the A C++/WinRT quick-start section, above).
- # The three main scenarios for C++/WinRT As you use and become familiar with C++/WinRT, and work through the rest of the
- documentation here, you'll likely notice that there are three main scenarios, as described
- in the following sections.
- # Consuming Windows APIs and types
- In other words, *using*, or *calling* APIs. For example, making API calls to communicate
- using Bluetooth; to stream and present video; to integrate with the Windows shell; and
- so on. C++/WinRT fully and uncompromisingly supports this category of scenario. For
- more info, see Consume APIs with C++/WinRT.
- # Authoring Windows APIs and types
- In other words, *producing* APIs and types. For example, producing the kinds of APIs
- described in the section above; or the graphics APIs; the storage and file system APIs;
- the networking APIs, and so on. For more info, see Author APIs with C++/WinRT.
- Authoring APIs with C++/WinRT is a little more involved than consuming them, because

- you must use IDL to define the shape of the API before you can implement it. There's a
- walkthrough of doing that in XAML controls; bind to a C++/WinRT property.

XAML applications

- This scenario is about building applications and controls on the XAML UI framework.
- Working in a XAML application amounts to a combination of consuming and authoring.
- But since XAML is the dominant UI framework on Windows today, and its influence over
- the Windows Runtime is proportionate to that, it deserves its own category of scenario.
- Be aware that XAML works best with programming languages that offer reflection. In
- C++/WinRT, you sometimes have to do a little extra work in order to interoperate with
- the XAML framework. All of those cases are covered in the documentation. Good places
- to start are XAML controls; bind to a C++/WinRT property and XAML custom
- (templated) controls with C++/WinRT.

Sample apps written in C++/WinRT

- See Where can I find C++/WinRT sample apps?.

Important APIsFeedback

- **Was this page helpful?**
- Provide product feedback
- | Get help at Microsoft Q&A
- SyndicationClient::RetrieveFeedAsync method
- SyndicationFeed.Items property
- winrt::hstring struct
- winrt::hresult-error struct
- C++/CX
- Error handling with C++/WinRT
- Interop between C++/WinRT and C++/CX
- Interop between C++/WinRT and the ABI
- Move to C++/WinRT from C++/CX
- String handling in C++/WinRT

Related topics

- **Yes**

NoFeedback

- **Was this page helpful?**
- Get help at Microsoft Q&A

Get Started with Win32 and C++

Article • 01/27/2022

The aim of this Get Started series is to teach you how to write a desktop program in C++ using Win32 and COM APIs. In the first module, you'll learn step-by-step how to create and show a window. Later modules will introduce the Component Object Model (COM), graphics and text, and user input.

For this series, it is assumed that you have a good working knowledge of C++ programming. No previous experience with Windows programming is assumed. If you are new to C++, learning material is available in the C++ language documentation .

Topic	Description
Intro to Win32 programming in C++	This section describes some of the basic terminology and coding conventions used in Windows programming.
Module 1. Your First Windows Program	In this module, you will create a simple Windows program that shows a blank window.
Module 2. Using COM in Your Windows Program	This module introduces the Component Object Model (COM), which underlies many of the modern Windows APIs.
Module 3. Windows Graphics	This module introduces the Windows graphics architecture, with a focus on Direct2D.
Module 4. User Input	This module describes mouse and keyboard input.
Sample Code	Contains links to download the sample code for this series.

In this section

 Yes

 No

Creating an MFC Application

Article • 02/14/2023

- An MFC application is an executable application for Windows that is based on the
- Microsoft Foundation Class (MFC) Library. MFC executables generally fall into five types:
- standard Windows applications, dialog boxes, forms-based applications, Explorer-style
- applications, and Web browser-style applications. For more information, see:
- Using the Classes to Write Windows Applications
- Creating and Displaying Dialog Boxes
- Creating a Forms-Based MFC Application
- Creating a File Explorer-Style MFC Application
- Creating a Web Browser-Style MFC Application
- The MFC Application Wizard generates the appropriate classes and files for any of these
- types of applications, depending on the options you select in the wizard.
- The easiest way to create an MFC application is to use the MFC Application Wizard (**MFC**
- **App project** in Visual Studio 2019). To create an MFC console application (a command
- line program that uses MFC libraries but runs in the console window), use the Windows
- Desktop Wizard and choose the **Console Application** and **MFC Headers** options.
- # To create an MFC forms or dialog-based
- # application
- 1. From the main menu, choose **File** > **New** > **Project**.
- 2. Enter "MFC" into the search box and then choose **MFC App** from the result list.
- 3. Modify the defaults as needed, then press **Create** to open the **MFC Application**
- **Wizard**.
- 4. Modify the configuration values as needed, then press **Finish**.
- For more information, see Creating a forms-based MFC application.# To create an MFC console application
- An MFC console application is a command-line program that uses MFC libraries but
- runs in the console window.
- 1. From the main menu, choose **File** > **New** > **Project**.

- 2. Enter "Desktop" into the search box and then choose **Windows Desktop Wizard**
- from the result list, then press **Next**.
- 3. Modify the project name and location as needed, then press **Create** to open the
- **Windows Desktop Wizard**.
- 4. Check the **MFC Headers** box and set other values as needed, then press **OK**.Once your project is created, you can view the files created in **Solution Explorer**. For
- more information about the files the wizard creates for your project, see the project
- generated file ReadMe.txt. For more information about the file types, see File Types
- Created for Visual Studio C++ projects.

See also

- Adding Functionality with Code Wizards
- Property Pages

Walkthrough: Create and use your own Dynamic Link Library (C++)

- Article • 12/10/2021
- This step-by-step walkthrough shows how to use the Visual Studio IDE to create your
- own dynamic link library (DLL) written in Microsoft C++ (MSVC). Then it shows how to
- use the DLL from another C++ app. DLLs (also known as *shared libraries* in UNIX-based
- operating systems) are one of the most useful kinds of Windows components. You can
- use them as a way to share code and resources, and to shrink the size of your apps.
- DLLs can even make it easier to service and extend your apps.
- In this walkthrough, you'll create a DLL that implements some math functions. Then
- you'll create a console app that uses the functions from the DLL. You'll also get an
- introduction to some of the programming techniques and conventions used in Windows
- DLLs.
- This walkthrough covers these tasks:
- Create a DLL project in Visual Studio.
- Add exported functions and variables to the DLL.
- Create a console app project in Visual Studio.

- Use the functions and variables imported from the DLL in the console app.
- Run the completed app.
- Like a statically linked library, a DLL *exports* variables, functions, and resources by name.
- A client app *imports* the names to use those variables, functions, and resources. Unlike a
- statically linked library, Windows connects the imports in your app to the exports in a
- DLL at load time or at run time, instead of connecting them at link time. Windows
- requires extra information that isn't part of the standard C++ compilation model to
- make these connections. The MSVC compiler implements some Microsoft-specific
- extensions to C++ to provide this extra information. We explain these extensions as we
- go.
- This walkthrough creates two Visual Studio solutions; one that builds the DLL, and one
- that builds the client app. The DLL uses the C calling convention. It can be called from
- apps written in other programming languages, as long as the platform, calling
- conventions, and linking conventions match. The client app uses *implicit linking*, whereWindows links the app to the DLL at load-time. This linking lets the app call the DLL
- supplied functions just like the functions in a statically linked library.
- This walkthrough doesn't cover some common situations. The code doesn't show the
- use of C++ DLLs by other programming languages. It doesn't show how to create a
- resource-only DLL, or how to use explicit linking to load DLLs at run-time rather than at
- load-time. Rest assured, you can use MSVC and Visual Studio to do all these things.
- Even though the code of the DLL is written in C++, we've used C-style interfaces for the
- exported functions. There are two main reasons for this: First, many other languages
- support imports of C-style functions. The client app doesn't have to be written in C++.
- Second, it avoids some common pitfalls related to exported classes and member
- functions. It's easy to make hard-to-diagnose errors when exporting classes, since

- everything referred to within a class declaration has to have an instantiation that's also
- exported. This restriction applies to DLLs, but not static libraries. If your classes are plain
- old-data style, you shouldn't run into this issue.
- For links to more information about DLLs, see Create C/C++ DLLs in Visual Studio. For
- more information about implicit linking and explicit linking, see Determine which linking
- method to use. For information about creating C++ DLLs for use with programming
- languages that use C-language linkage conventions, see Exporting C++ functions for
- use in C-language executables. For information about how to create DLLs for use with
- .NET languages, see Calling DLL Functions from Visual Basic Applications.

Prerequisites

- A computer that runs Microsoft Windows 7 or later versions. We recommend the
- latest version of Windows for the best development experience.
- A copy of Visual Studio. For information on how to download and install Visual
- Studio, see Install Visual Studio. When you run the installer, make sure that the
- **Desktop development with C++** workload is checked. Don't worry if you didn't
- install this workload when you installed Visual Studio. You can run the installer
- again and install it now.
- An understanding of the basics of using the Visual Studio IDE. If you've used
- Windows desktop apps before, you can probably keep up. For an introduction, seeVisual Studio IDE feature tour.
- An understanding of enough of the fundamentals of the C++ language to follow
- along. Don't worry, we don't do anything too complicated.

Create the DLL project

- In this set of tasks, you create a project for your DLL, add code, and build it. To begin,
- start the Visual Studio IDE, and sign in if you need to. The instructions vary slightly
- depending on which version of Visual Studio you're using. Make sure you have the

- correct version selected in the control in the upper left of this page.
- # To create a DLL project in Visual Studio 2019
- 1. On the menu bar, choose **File** > **New** > **Project** to open the **Create a New Project**
- dialog box.
- 2. At the top of the dialog, set **Language** to **C++**, set **Platform** to **Windows**, and set
- **Project type** to **Library**.
- 3. From the filtered list of project types, select **Dynamic-link Library (DLL)**, and then
- choose **Next**.
- 4. In the **Configure your new project** page, enter *MathLibrary* in the **Project name**
- box to specify a name for the project. Leave the default **Location** and **Solution**
- **name** values. Set **Solution** to **Create new solution**. Uncheck **Place solution and**
- **project in the same directory** if it's checked.
- 5. Choose the **Create** button to create the project.When the solution is created, you can see the generated project and source files in the
- **Solution Explorer** window in Visual Studio.
- Right now, this DLL doesn't do very much. Next, you'll create a header file to declare the
- functions your DLL exports, and then add the function definitions to the DLL to make it
- more useful.
- # To add a header file to the DLL
- 1. To create a header file for your functions, on the menu bar, choose **Project** > **Add**
- **New Item**.
- 2. In the **Add New Item** dialog box, in the left pane, select **Visual C++**. In the center
- pane, select **Header File (.h)**. Specify *MathLibrary.h* as the name for the header file.3. Choose the **Add** button to generate a blank header file, which is displayed in a new
- editor window.
- 4. Replace the contents of the header file with this code:
- C++
- ```
  // MathLibrary.h - Contains declarations of math functions
  #pragma once
  #ifdef MATHLIBRARY_EXPORTS
  #define MATHLIBRARY_API __declspec(dllexport)
  #else
  #define MATHLIBRARY_API __declspec(dllimport)
  ```
  ```

- `#endif`
- `// The Fibonacci recurrence relation describes a sequence F`
- `// where F(n) is { n = 0, a`
- `// { n = 1, b`This header file declares some functions to produce a generalized Fibonacci sequence,
- given two initial values. A call to `fibonacci_init(1, 1)` generates the familiar Fibonacci
- number sequence.
- Notice the preprocessor statements at the top of the file. The new project template for a
- DLL project adds `<PROJECTNAME>_EXPORTS` to the defined preprocessor macros. In this
- example, Visual Studio defines `MATHLIBRARY_EXPORTS` when your MathLibrary DLL project
- is built.
- When the `MATHLIBRARY_EXPORTS` macro is defined, the `MATHLIBRARY_API` macro sets the
- `__declspec(dllexport)` modifier on the function declarations. This modifier tells the
- compiler and linker to export a function or variable from the DLL for use by other
- applications. When `MATHLIBRARY_EXPORTS` is undefined, for example, when the header file
- is included by a client application, `MATHLIBRARY_API` applies the `__declspec(dllimport)`
- modifier to the declarations. This modifier optimizes the import of the function or
- variable in an application. For more information, see dllexport, dllimport.
- 1. In **Solution Explorer**, right-click on the **Source Files** node and choose **Add** > **New**
- **Item**. Create a new .cpp file called *MathLibrary.cpp*, in the same way that you
- added a new header file in the previous step.
- `// { n > 1, F(n-2) + F(n-1)`
- `// for some initial integral values a and b.`
- `// If the sequence is initialized F(0) = 1, F(1) = 1,`
- `// then this relation produces the well-known Fibonacci`
- `// sequence: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...`
- `// Initialize a Fibonacci relation sequence`
- `// such that F(0) = a, F(1) = b.`
- `// This function must be called before any other function.`
- `extern "C" MATHLIBRARY_API void fibonacci_init(`
- `const unsigned long long a, const unsigned long long b);`
- `// Produce the next value in the sequence.`
- `// Returns true on success and updates current value and index;`
- `// false on overflow, leaves current value and index unchanged.`
- `extern "C" MATHLIBRARY_API bool fibonacci_next();`

- `// Get the current value in the sequence.`
- `extern "C" MATHLIBRARY_API unsigned long long fibonacci_current();`
- `// Get the position of the current value in the sequence.`
- `extern "C" MATHLIBRARY_API unsigned fibonacci_index();`
- **To add an implementation to the DLL**2. In the editor window, select the tab for **MathLibrary.cpp** if it's already open. If not,
- in **Solution Explorer**, double-click **MathLibrary.cpp** in the **Source Files** folder of the
- **MathLibrary** project to open it.
- 3. In the editor, replace the contents of the MathLibrary.cpp file with the following
- code:
- C++
- `// MathLibrary.cpp : Defines the exported functions for the DLL.`
- `#include "pch.h" // use stdafx.h in Visual Studio 2017 and earlier`
- `#include <utility>`
- `#include <limits.h>`
- `#include "MathLibrary.h"`
- `// DLL internal state variables:`
- `static unsigned long long previous_; // Previous value, if any`
- `static unsigned long long current_; // Current sequence value`
- `static unsigned index_; // Current seq. position`
- `// Initialize a Fibonacci relation sequence`
- `// such that F(0) = a, F(1) = b.`
- `// This function must be called before any other function.`
- `void fibonacci_init(`
- `const unsigned long long a,`
- `const unsigned long long b)`
- `{`
- `index_ = 0;`
- `current_ = a;`
- `previous_ = b; // see special case when initialized`
- `}`
- `// Produce the next value in the sequence.`
- `// Returns true on success, false on overflow.`
- `bool fibonacci_next()`
- `{`
- `// check to see if we'd overflow result or position`
- `if ((ULLONG_MAX - previous_ < current_) ||`
- `(UINT_MAX == index_))`
- `{`
- `return false;`
- `}`
- `// Special case when index == 0, just return b value`
- `if (index_ > 0)`
- `{`
- `// otherwise, calculate next sequence value`
- `previous_ += current_;`
- `}`
- `std::swap(current_, previous_);`

- ++index_;
- return true;
- }To verify that everything works so far, compile the dynamic link library. To compile,
- choose **Build** > **Build Solution** on the menu bar. The DLL and related compiler output
- are placed in a folder called *Debug* directly below the solution folder. If you create a
- Release build, the output is placed in a folder called *Release*. The output should look
- something like this:
- Output
- Congratulations, you've created a DLL using Visual Studio! Next, you'll create a client
- app that uses the functions exported by the DLL.
- When you create a DLL, think about how client apps may use it. To call the functions or
- access the data exported by a DLL, client source code must have the declarations
- available at compile time. At link time, the linker requires information to resolve the
- function calls or data accesses. A DLL supplies this information in an *import library*, a file
- that contains information about how to find the functions and data, instead of the actual
- code. And at run time, the DLL must be available to the client, in a location that the
- operating system can find.
- // Get the current value in the sequence.
- unsigned long long fibonacci_current()
- {
- return current_;
- }
- // Get the current index position in the sequence.
- unsigned fibonacci_index()
- {
- return index_;
- }
- 1>------ Build started: Project: MathLibrary, Configuration: Debug Win32 ---
- ---
- 1>pch.cpp
- 1>dllmain.cpp
- 1>MathLibrary.cpp
- 1>Generating Code...
- 1> Creating library
- C:\Users\username\Source\Repos\MathLibrary\Debug\MathLibrary.lib and object
- C:\Users\username\Source\Repos\MathLibrary\Debug\MathLibrary.exp

- `1>MathLibrary.vcxproj ->`
- `C:\Users\username\Source\Repos\MathLibrary\Debug\MathLibrary.dll`
- `========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========`
- # Create a client app that uses the DLL

  Whether it's your own or from a third-party, your client app project needs several pieces
- of information to use a DLL. It needs to find the headers that declare the DLL exports,
- the import libraries for the linker, and the DLL itself. One solution is to copy all of these
- files into your client project. For third-party DLLs that are unlikely to change while your
- client is in development, this method may be the best way to use them. However, when
- you also build the DLL, it's better to avoid duplication. If you make a local copy of DLL
- files that are under development, you may accidentally change a header file in one copy
- but not the other, or use an out-of-date library.
- To avoid out-of-sync code, we recommend you set the include path in your client
- project to include the DLL header files directly from your DLL project. Also, set the
- library path in your client project to include the DLL import libraries from the DLL
- project. And finally, copy the built DLL from the DLL project into your client build output
- directory. This step allows your client app to use the same DLL code you build.
- # To create a client app in Visual Studio
- 1. On the menu bar, choose **File** > **New** > **Project** to open the **Create a new project**
- dialog box.
- 2. At the top of the dialog, set **Language** to **C++**, set **Platform** to **Windows**, and set
- **Project type** to **Console**.
- 3. From the filtered list of project types, choose **Console App** then choose **Next**.
- 4. In the **Configure your new project** page, enter *MathClient* in the **Project name** box
- to specify a name for the project. Leave the default **Location** and **Solution name**

- values. Set **Solution** to **Create new solution**. Uncheck **Place solution and project in**
- **the same directory** if it's checked.
- 5. Choose the **Create** button to create the client project.A minimal console application project is created for you. The name for the main source
- file is the same as the project name that you entered earlier. In this example, it's named
- **MathClient.cpp**. You can build it, but it doesn't use your DLL yet.
- Next, to call the MathLibrary functions in your source code, your project must include
- the *MathLibrary.h* file. You could copy this header file into your client app project, then
- add it to the project as an existing item. This method can be a good choice for third
- party libraries. However, if you're working on the code for your DLL and your client at
- the same time, the header files could get out of sync. To avoid this issue, set the
- **Additional Include Directories** path in your project to include the path to the original
- header.
- # To add the DLL header to your include path
- 1. Right-click on the **MathClient** node in **Solution Explorer** to open the **Property**
- **Pages** dialog.
- 2. In the **Configuration** drop-down box, select **All Configurations** if it's not already
- selected.
- 3. In the left pane, select **Configuration Properties** > **C/C++** > **General**.
- 4. In the property pane, select the drop-down control next to the **Additional Include**
- **Directories** edit box, and then choose **Edit**.5. Double-click in the top pane of the **Additional Include Directories** dialog box to
- enable an edit control. Or, choose the folder icon to create a new entry.
- 6. In the edit control, specify the path to the location of the **MathLibrary.h** header
- file. You can choose the ellipsis (**...**) control to browse to the correct folder.
- You can also enter a relative path from your client source files to the folder that

- contains the DLL header files. If you followed the directions to put your client
- project in a separate solution from the DLL, the relative path should look like this:
- `..\..\MathLibrary\MathLibrary`
- If your DLL and client projects are in the same solution, the relative path might
- look like this:
- `..\MathLibrary`
- When the DLL and client projects are in other folders, adjust the relative path to
- match. Or, use the ellipsis control to browse for the folder.
- 7. After you've entered the path to the header file in the **Additional Include**
- **Directories** dialog box, choose the **OK** button. In the **Property Pages** dialog box,
- choose the **OK** button to save your changes.
- You can now include the **MathLibrary.h** file and use the functions it declares in your
- client application. Replace the contents of **MathClient.cpp** by using this code:
- C++This code can be compiled, but not linked. If you build the client app now, the error list
- shows several LNK2019 errors. That's because your project is missing some information:
- You haven't specified that your project has a dependency on the *MathLibrary.lib* library
- yet. And, you haven't told the linker how to find the *MathLibrary.lib* file.
- To fix this issue, you could copy the library file directly into your client app project. The
- linker would find and use it automatically. However, if both the library and the client app
- are under development, that might lead to changes in one copy that aren't shown in the
- other. To avoid this issue, you can set the **Additional Dependencies** property to tell the
- build system that your project depends on *MathLibrary.lib*. And, you can set an
- **Additional Library Directories** path in your project to include the path to the original
- library when you link.
- 1. Right-click on the **MathClient** node in **Solution Explorer** and choose **Properties** to
- open the **Property Pages** dialog.
- 2. In the **Configuration** drop-down box, select **All Configurations** if it's not already

- selected. It ensures that any property changes apply to both Debug and Release
- builds.
- 3. In the left pane, select **Configuration Properties** > **Linker** > **Input**. In the property
- pane, select the drop-down control next to the **Additional Dependencies** edit box,
- and then choose **Edit**.
- `// MathClient.cpp : Client app for MathLibrary DLL.`
- `// #include "pch.h" Uncomment for Visual Studio 2017 and earlier`
- `#include <iostream>`
- `#include "MathLibrary.h"`
- `int main()`
- `{`
- `// Initialize a Fibonacci relation sequence.`
- `fibonacci_init(1, 1);`
- `// Write out the sequence values until overflow.`
- `do {`
- `std::cout << fibonacci_index() << ": "`
- `<< fibonacci_current() << std::endl;`
- `} while (fibonacci_next());`
- `// Report count of values written before overflow.`
- `std::cout << fibonacci_index() + 1 <<`
- `" Fibonacci sequence values fit in an " <<`
- `"unsigned 64-bit integer." << std::endl;`
- `}`
- **To add the DLL import library to your project**4. In the **Additional Dependencies** dialog, add *MathLibrary.lib* to the list in the top
- edit control.
- 5. Choose **OK** to go back to the **Property Pages** dialog box.
- 6. In the left pane, select **Configuration Properties** > **Linker** > **General**. In the
- property pane, select the drop-down control next to the **Additional Library**
- **Directories** edit box, and then choose **Edit**.7. Double-click in the top pane of the **Additional Library Directories** dialog box to
- enable an edit control. In the edit control, specify the path to the location of the
- **MathLibrary.lib** file. By default, it's in a folder called *Debug* directly under the DLL
- solution folder. If you create a release build, the file is placed in a folder called
- *Release*. You can use the `$(IntDir)` macro so that the linker can find your DLL, no
- matter which kind of build you create. If you followed the directions to put your

- client project in a separate solution from the DLL project, the relative path should
- look like this:
- `..\..\MathLibrary\$(IntDir)`
- If your DLL and client projects are in other locations, adjust the relative path to
- match.8. Once you've entered the path to the library file in the **Additional Library**
- **Directories** dialog box, choose the **OK** button to go back to the **Property Pages**
- dialog box. Choose **OK** to save the property changes.
- Your client app can now compile and link successfully, but it still doesn't have everything
- it needs to run. When the operating system loads your app, it looks for the MathLibrary
- DLL. If it can't find the DLL in certain system directories, the environment path, or the
- local app directory, the load fails. Depending on the operating system, you'll see an
- error message like this:
- One way to avoid this issue is to copy the DLL to the directory that contains your client
- executable as part of the build process. You can add a **Post-Build Event** to your project,
- to add a command that copies the DLL to your build output directory. The command
- specified here copies the DLL only if it's missing or has changed. It uses macros to copy
- to and from the Debug or Release locations, based on your build configuration.
- # To copy the DLL in a post-build event
- 1. Right-click on the **MathClient** node in **Solution Explorer** and choose **Properties** to
- open the **Property Pages** dialog.2. In the **Configuration** drop-down box, select **All Configurations** if it isn't already
- selected.
- 3. In the left pane, select **Configuration Properties** > **Build Events** > **Post-Build**
- **Event**.
- 4. In the property pane, select the edit control in the **Command Line** field. If you
- followed the directions to put your client project in a separate solution from the
- DLL project, then enter this command:
- `xcopy /y /d "..\..\MathLibrary\$(IntDir)MathLibrary.dll" "$(OutDir)"`

- If your DLL and client projects are in other directories, change the relative path to
- the DLL to match.
- 5. Choose the **OK** button to save your changes to the project properties.
- Now your client app has everything it needs to build and run. Build the application by
- choosing **Build** > **Build Solution** on the menu bar. The **Output** window in Visual Studio
- should have something like the following example depending on your version of Visual
- Studio:
- Output
- `1>------ Build started: Project: MathClient, Configuration: Debug Win32 ----`
- `--`
- `1>MathClient.cpp`
- `1>MathClient.vcxproj ->C:\Users\username\Source\Repos\MathClient\Debug\MathClient.exe`
- `1>1 File(s) copied`
- `========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========`
- Congratulations, you've created an application that calls functions in your DLL. Now run
- your application to see what it does. On the menu bar, choose **Debug** > **Start Without**
- **Debugging**. Visual Studio opens a command window for the program to run in. The last
- part of the output should look like:
- Press any key to dismiss the command window.
- Now that you've created a DLL and a client application, you can experiment. Try setting
- breakpoints in the code of the client app, and run the app in the debugger. See what
- happens when you step into a library call. Add other functions to the library, or write
- another client app that uses your DLL.
- When you deploy your app, you must also deploy the DLLs it uses. The simplest way to
- make the DLLs that you build, or that you include from third parties, available is to put
- them in the same directory as your app. It's known as *app-local deployment*. For more
- information about deployment, see Deployment in Visual C++.
- # See also

- Calling DLL Functions from Visual Basic Applications

Walkthrough: Create and use a static library

- Article • 10/29/2021

This step-by-step walkthrough shows how to create a static library (.lib file) for use with C++ apps. Using a static library is a great way to reuse code. Rather than reimplementing the same routines in every app that requires the functionality, you write them one time in a static library and then reference it from the apps. Code linked from a static library becomes part of your app—you don't have to install another file to use the code.

This walkthrough covers these tasks:

- Create a static library project
- Add a class to the static library
- Create a C++ console app that references the static library
- Use the functionality from the static library in the app
- Run the app

Prerequisites

An understanding of the fundamentals of the C++ language.

Create a static library project

The instructions for how to create the project vary depending on your version of Visual Studio. To see the documentation for your preferred version of Visual Studio, use the **Version** selector control. It's found at the top of the table of contents on this page.

To create a static library project in Visual Studio

1. On the menu bar, choose **File** > **New** > **Project** to open the **Create a New Project** dialog.
2. At the top of the dialog, set **Language** to **C++**, set **Platform** to **Windows**, and set **Project type** to **Library**.3. From the filtered list of project types, select **Windows Desktop Wizard**, then

- choose **Next**.
- 4. In the **Configure your new project** page, enter *MathLibrary* in the **Project name**
- box to specify a name for the project. Enter *StaticMath* in the **Solution name** box.
- Choose the **Create** button to open the **Windows Desktop Project** dialog.
- 5. In the **Windows Desktop Project** dialog, under **Application type**, select **Static**
- **Library (.lib)**.
- 6. Under **Additional options**, uncheck the **Precompiled header** check box if it's
- checked. Check the **Empty project** box.
- 7. Choose **OK** to create the project.
- 1. To create a header file for a new class, right-click to open the shortcut menu for
- the **MathLibrary** project in **Solution Explorer**, and then choose **Add** > **New Item**.
- 2. In the **Add New Item** dialog box, select **Visual C++** > **Code**. In the center pane,
- select **Header File (.h)**. Specify a name for the header file—for example,
- *MathLibrary.h*—and then choose the **Add** button. A nearly blank header file is
- displayed.
- 3. Add a declaration for a class named `Arithmetic` to do common mathematical
- operations such as addition, subtraction, multiplication, and division. The code
- should resemble:
- C++
- # Add a class to the static library
- ## To add a class to the static library
- ```
  // MathLibrary.h
  #pragma once
  namespace MathLibrary
  {
  class Arithmetic
  {
  public:
  // Returns a + b
  static double Add(double a, double b);
  // Returns a - b
  static double Subtract(double a, double b);
  ```
  4. To create a source file for the new class, open the shortcut menu for the
  ```

- **MathLibrary** project in **Solution Explorer**, and then choose **Add** > **New Item**.
- 5. In the **Add New Item** dialog box, in the center pane, select **C++ File (.cpp)**. Specify
- a name for the source file—for example, *MathLibrary.cpp*—and then choose the
- **Add** button. A blank source file is displayed.
- 6. Use this source file to implement the functionality for class `Arithmetic` . The code
- should resemble:
- C++
- `// Returns a * b`
- `static double Multiply(double a, double b);`
- `// Returns a / b`
- `static double Divide(double a, double b);`
- `};`
- `}`
- `// MathLibrary.cpp`
- `// compile with: cl /c /EHsc MathLibrary.cpp`
- `// post-build command: lib MathLibrary.obj`
- `#include "MathLibrary.h"`
- `namespace MathLibrary`
- `{`
- `double Arithmetic::Add(double a, double b)`
- `{`
- `return a + b;`
- `}`
- `double Arithmetic::Subtract(double a, double b)`
- `{`
- `return a - b;`
- `}`
- `double Arithmetic::Multiply(double a, double b)`
- `{`
- `return a * b;`
- `}`
- `double Arithmetic::Divide(double a, double b)`
- `{`
- `return a / b;`
- `}`
- }7. To build the static library, select **Build** > **Build Solution** on the menu bar. The build
- creates a static library, *MathLibrary.lib*, that can be used by other programs.
- **Note**
- When you build on the Visual Studio command line, you must build the
- program in two steps. First, run `cl /c /EHsc MathLibrary.cpp` to compile the
- code and create an object file that's named *MathLibrary.obj*. (The `cl`

- command invokes the compiler, Cl.exe, and the `/c` option specifies compile
- without linking. For more information, see **/c (Compile Without Linking)**.)
- Second, run `lib MathLibrary.obj` to link the code and create the static library
- *MathLibrary.lib*. (The `lib` command invokes the Library Manager, Lib.exe. For
- more information, see **LIB Reference**.)

Create a C++ console app that references the static library

To create a C++ console app that references the static library in Visual Studio

- 1. In **Solution Explorer**, right-click on the top node, **Solution 'StaticMath'**, to open
- the shortcut menu. Choose **Add** > **New Project** to open the **Add a New Project**
- dialog.
- 2. At the top of the dialog, set the **Project type** filter to **Console**.
- 3. From the filtered list of project types, choose **Console App** then choose **Next**. In
- the next page, enter *MathClient* in the **Name** box to specify a name for the project.
- 4. Choose the **Create** button to create the client project.
- 5. After you create a console app, an empty program is created for you. The name for
- the source file is the same as the name that you chose earlier. In the example, it's
- named `MathClient.cpp` .

Use the functionality from the static library in the app

To use the functionality from the static library in the app

- 1. Before you can use the math routines in the static library, you must reference it.
- Open the shortcut menu for the **MathClient** project in **Solution Explorer**, and then

- choose **Add** > **Reference**.
- 2. The **Add Reference** dialog box lists the libraries that you can reference. The
- **Projects** tab lists the projects in the current solution and any libraries they
- reference. Open the **Projects** tab, select the **MathLibrary** check box, and then
- choose the **OK** button.
- 3. To reference the `MathLibrary.h` header file, you must modify the included
- directories path. In **Solution Explorer**, right-click on **MathClient** to open the
- shortcut menu. Choose **Properties** to open the **MathClient Property Pages** dialog
- box.
- 4. In the **MathClient Property Pages** dialog box, set the **Configuration** drop-down to
- **All Configurations**. Set the **Platform** drop-down to **All Platforms**.
- 5. Select the **Configuration Properties** > **C/C++** > **General** property page. In the
- **Additional Include Directories** property, specify the path of the **MathLibrary**
- directory, or browse for it.
- To browse for the directory path:
- a. Open the **Additional Include Directories** property value drop-down list, and
- then choose **Edit**.
- b. In the **Additional Include Directories** dialog box, double-click in the top of the
- text box. Then choose the ellipsis button (**...**) at the end of the line.
- c. In the **Select Directory** dialog box, navigate up a level, and then select the
- **MathLibrary** directory. Then choose the **Select Folder** button to save your
- selection.
- d. In the **Additional Include Directories** dialog box, choose the **OK** button.
- e. In the **Property Pages** dialog box, choose the **OK** button to save your changes
- to the project.
- 6. You can now use the `Arithmetic` class in this app by including the `#include`
- `"MathLibrary.h"` header in your code. Replace the contents of `MathClient.cpp` with
- this code:

- C++7. To build the executable, choose **Build** > **Build Solution** on the menu bar.
- 1. Make sure that **MathClient** is selected as the default project. To select it, right-click
- to open the shortcut menu for **MathClient** in **Solution Explorer**, and then choose
- **Set as StartUp Project**.
- 2. To run the project, on the menu bar, choose **Debug** > **Start Without Debugging**.
- The output should resemble:
- Output
- ```
  // MathClient.cpp
  // compile with: cl /EHsc MathClient.cpp /link MathLibrary.lib
  #include <iostream>
  #include "MathLibrary.h"
  int main()
  {
  double a = 7.4;
  int b = 99;
  std::cout << "a + b = " <<
  MathLibrary::Arithmetic::Add(a, b) << std::endl;
  std::cout << "a - b = " <<
  MathLibrary::Arithmetic::Subtract(a, b) << std::endl;
  std::cout << "a * b = " <<
  MathLibrary::Arithmetic::Multiply(a, b) << std::endl;
  std::cout << "a / b = " <<
  MathLibrary::Arithmetic::Divide(a, b) << std::endl;
  return 0;
  }
  ```
- # Run the app
- # To run the app
- ```
  a + b = 106.4
  a - b = -91.6
  a * b = 732.6
  a / b = 0.0747475
  ```
- # See alsoWalkthrough: Creating and Using a Dynamic Link Library (C++)
- Desktop Applications (Visual C++)# Walkthrough: Compile a C++/CLI
- # program that targets the CLR in Visual

Studio

Article • 10/29/2021

By using C++/CLI you can create C++ programs that use .NET classes as well as native C++ types. C++/CLI is intended for use in console applications and in DLLs that wrap native C++ code and make it accessible from .NET programs. To create a Windows user interface based on .NET, use C# or Visual Basic.

For this procedure, you can type your own C++ program or use one of the sample programs. The sample program that we use in this procedure creates a text file named textfile.txt, and saves it to the project directory.

Prerequisites

An understanding of the fundamentals of the C++ language.

In Visual Studio 2017 and later, C++/CLI support is an optional component. To install it, open the **Visual Studio Installer** from the Windows Start menu. Make sure that the **Desktop development with C++** tile is checked, and in the **Optional** components section, also check **C++/CLI Support**.

Create a new project

The following steps vary depending on which version of Visual Studio you are using. To see the documentation for your preferred version of Visual Studio, use the **Version** selector control. It's found at the top of the table of contents on this page.

To create a C++/CLI project in Visual Studio

1. In **Solution Explorer**, right-click on the top to open the **Create a New Project** dialog box.
2. At the top of the dialog, type **CLR** in the search box and then choose **CLR Empty Project** from the results list.
3. Choose the **Create** button to create the project.

Add a source file

- 1. If **Solution Explorer** isn't visible, click **Solution Explorer** on the **View** menu.
- 2. Add a new source file to the project:
- Right-click the **Source Files** folder in **Solution Explorer**, point to **Add**, and
- click **New Item**.
- Click **C++ File (.cpp)** and type a file name and then click **Add**.
- The **.cpp** file appears in the **Source Files** folder in **Solution Explorer** and a tabbed
- window appears where you type the code you want in that file.
- 3. Click in the newly created tab in Visual Studio and type a valid Visual C++
- program, or copy and paste one of the sample programs.
- For example, you can use the How to: Write a Text File (C++/CLI) sample program
- (in the **File Handling and I/O** node of the Programming Guide).
- If you use the sample program, notice that you use the **gcnew** keyword instead of
- **new** when creating a .NET object, and that **gcnew** returns a handle (`^`) rather than a
- pointer (`*`):
- `StreamWriter^ sw = gcnew StreamWriter(fileName);`
- For more information on C++/CLI syntax, see Component Extensions for Runtime
- Platforms.
- 4. On the **Build** menu, click **Build Solution**.
- The **Output** window displays information about the compilation progress, such as
- the location of the build log and a message that indicates the build status.
- If you make changes and run the program without doing a build, a dialog box
- might indicate that the project is out of date. Select the checkbox on this dialog
- before you click **OK** if you want Visual Studio to always use the current versions of
- files instead of prompting you each time it builds the application.
- 5. On the **Debug** menu, click **Start without Debugging**.
- 6. If you used the sample program, when you run the program a command window is
- displayed that indicates the text file has been created.
- The **textfile.txt** text file is now located in your project directory. You can open this
- file by using Notepad. **Note**
- Choosing the empty CLR project template automatically set the `/clr`

- compiler option. To verify this, right-click the project in **Solution Explorer** and
- clicking **Properties**, and then check the **Common Language Runtime support**
- option in the **General** node of **Configuration Properties**.

See also

- C++ Language Reference
- Projects and build systems

Create a simple Universal Windows Platform (UWP) game with DirectX

- Article • 10/20/2022
- In this set of tutorials, you'll learn how to use DirectX and C++/WinRT to create the basic
- Universal Windows Platform (UWP) sample game named **Simple3DGameDX**. The
- gameplay takes place in a simple first-person 3D shooting gallery.
- **Note**
- The link from which you can download the **Simple3DGameDX** sample game itself is
- **Direct3D sample game**. The C++/WinRT source code is in the folder named
- `cppwinrt` . For info about other UWP sample apps, see **Sample applications for**
- **Windows development**.
- These tutorials cover all of the major parts of a game, including the processes for
- loading assets such as arts and meshes, creating a main game loop, implementing a
- simple rendering pipeline, and adding sound and controls.
- You'll also see UWP game development techniques and considerations. We'll focus on
- key UWP DirectX game development concepts, and call out Windows-Runtime-specific
- considerations around those concepts.

Objective

- To learn about the basic concepts and components of a UWP DirectX game, and to
- become more comfortable designing UWP games with DirectX.

- # What you need to know
- For this tutorial, you need to be familiar with these subjects.
- C++/WinRT. C++/WinRT is a standard modern C++17 language projection for
- Windows APIs, implemented as a header-file-based library, and designed to
- provide you with first-class access to the modern Windows APIs.
- Basic linear algebra and Newtonian physics concepts.
- Basic graphics programming terminology.
- Basic Windows programming concepts.
- Basic familiarity with the Direct2D and Direct3D 11 APIs.The **Simple3DGameDX** sample game implements a simple first-person 3D shooting
- gallery, where the player fires balls at moving targets. Hitting each target awards a set
- number of points, and the player can progress through 6 levels of increasing challenge.
- At the end of the levels, the points are tallied, and the player is awarded a final score.
- The sample demonstrates these game concepts.
- Interoperation between DirectX 11.1 and the Windows Runtime
- A first-person 3D perspective and camera
- Stereoscopic 3D effects
- Collision-detection between objects in 3D
- Handling player input for mouse, touch, and Xbox controller controls
- Audio mixing and playback
- A basic game state-machine
- **Topic**
- **Description**
- Set up the
- game project
- The first step in developing your game is to set up a project in Microsoft Visual
- Studio. After you've configured a project specifically for game development, you
- could later re-use it as a kind of template.
- Define the
- game's UWP
- app framework
- The first step in coding a Universal Windows Platform (UWP) game is building
- the framework that lets the app object interact with Windows.
- Game flow
- management
- Define the high-level state machine to enable player and system interaction.
- Learn how UI interacts with the overall game's state machine and how to create
- event handlers for UWP games.
- # Direct3D UWP shooting gallery sample**Topic**

- **Description**
- Define the
- main game
- object
- Now, we look at the details of the sample game's main object and how the rules
- it implements translate into interactions with the game world.
- Rendering
- framework I:
- Intro to
- rendering
- Learn how to develop the rendering pipeline to display graphics. Intro to
- rendering.
- Rendering
- framework II:
- Game
- rendering
- Learn how to assemble the rendering pipeline to display graphics. Game
- rendering, set up and prepare data.
- Add a user
- interface
- Learn how to add a 2D user interface overlay to a DirectX UWP game.
- Add controls
- Now, we take a look at how the sample game implements move-look controls
- in a 3-D game, and how to develop basic touch, mouse, and game controller
- controls.
- Add sound
- Develop a simple sound engine using XAudio2 APIs to playback game music
- and sound effects.
- Extend the
- sample game
- Learn how to implement a XAML overlay for a UWP DirectX
  game.# Tutorial: Open a project from a repo
- Article • 12/19/2024
- In this tutorial, you use Visual Studio to connect to a repository, or *repo*, for the first
- time, clone it, and then open a project from it.
- In this tutorial, you learn how to:
- Open a project from a GitHub repo
- Browse to an Azure DevOps repo
- # Prerequisites
- If you don't have Visual Studio yet, go to Visual Studio downloads to install it for
- free.
- # Open a project from a GitHub repo

- Visual Studio makes it easy to open a project from a repo. You can do so when you start
- Visual Studio, or you can do so directly from within the Visual Studio IDE.
- Here's how.

Use the start window

- 1. Open Visual Studio.
- 2. On the start window, select **Clone a repository**.
- 3. Enter or type the repository location, and then select **Clone**.
-
- 4. If you're not already signed in, you might be prompted to sign into Visual Studio
- or your GitHub account.
- **Tip**
- For more information about signing in to Visual Studio, see **Sign in or switch**
- **Visual Studio user accounts**. For specific information about how to use yourGitHub account to sign in, see **Add your GitHub accounts to your Visual**
- **Studio keychain**. You might receive a trust notification. For more information,
- see **Configure trust settings for files and folders**.

View files in Solution Explorer

- Visual Studio loads the solutions from the repository by using the **Folder View** in
- **Solution Explorer**.
- You can view a solution in **Solution View** by double-clicking its `.sln` file.
- You can select **Switch Views** to switch between folder view and solution view.
- **Tip**You can change from the default Folder View to Solution View from the **Git** menu.
- Select **Settings** > **Source Control** > **Git Global Settings** > **Automatically load the**
- **solution when opening a Git repository**.

Open a project locally from a previously cloned GitHub repo

- 1. Open Visual Studio.
- 2. On the start window, select **Open a project or solution**.
- Visual Studio opens an instance of File Explorer, where you can browse to your
- solution or project, and then select it to open it.
- **Tip**

- If you opened the project or solution recently, select it from the **Open recent**
- section.
- Start coding!

Use the IDE

- You can also use the **Git** menu or the **Select Repository** control in the Visual Studio IDE
- to interact with a repository's folders and files.Here's how.

To clone a repo and open a project

- 1. In the Visual Studio IDE, select the **Git** menu, and then select **Clone Repository**.
- 2. Follow the prompts to connect to the Git repository that includes the files that
- you're looking for.

To open local folders and files

- 1. In the Visual Studio IDE, select the **Git** menu, select **Local Repositories**, and then
- select **Open Local Repository**.
- 2. Follow the prompts to connect to the Git repository that has the files that you're
- looking for.

Browse to an Azure DevOps repo

- Here's how to browse to and clone an Azure DevOps repo by using Visual Studio.
- 1. Open Visual Studio.
- 2. On the start window, select **Clone a repository**.
- 3. In the **Browse a repository** section, select **Azure DevOps**.
- 4. Follow the prompts to clone an Azure DevOps repo that includes the files that
- you're looking for, and then open your project.

Related content

- Feel free to dive into any of the following language-specific tutorials:

Feedback

- **Was this page helpful?**
- Provide product feedback
- | Ask the community
- Visual Studio tutorials | **C#**
- Visual Studio tutorials | **Visual Basic**
- Visual Studio tutorials | **C++**
- Visual Studio tutorials | **Python**
- Visual Studio tutorials | **JavaScript**, **TypeScript**, and **Node.js**
- For more information, see:
- About Git in Visual Studio

- Brpwse a repo
- Manage a repo
- Yes
- No# Learn to use the code editor
- Article • 01/24/2025
- In this 10-minute introduction to the code editor in Visual Studio, we'll add code to a file
- to look at some of the ways that Visual Studio makes writing, navigating, and
- understanding code easier.
- If you haven't already installed Visual Studio, go to the Visual Studio downloads page
- to install it for free.
- This article assumes you're already familiar with a programming language. If you aren't,
- we suggest you look at one of the programming quickstarts first, such as create a web
- app with Python or C#, or create a console app with Visual Basic or C++.
- **Tip**
- To follow along with this article, make sure you have the C# settings selected for
- Visual Studio. For information about selecting settings for the integrated
- development environment (IDE), see **Select environment settings**.
- # Create a new code file
- Start by creating a new file and adding some code to it.
- 1. Open Visual Studio. Select the **Esc** key, or select **Continue without code** on the
- start window, to open the development environment.
- 2. From the **File** menu on the menu bar, select **New** > **File**, or select the **Ctrl**+**N** keys.
- 3. In the **New File** dialog box, under the **General** category, select **C# Class**, and then
- select **Open**.
- A new file opens in the editor with the skeleton of a C# class.# Use GitHub Copilot
- GitHub Copilot acts as an AI pair programmer to provide autocomplete-style code

- completions and context-aware multi-line code suggestions, as you code, in real-time,
- right in the editor. GitHub Copilot turns natural language prompts including comments
- and method names into coding suggestions. You can view and incorporate suggestions
- from GitHub Copilot directly within the editor.
-
- Let's use Copilot to generate code suggestions:
- 1. Place your cursor just below the final closing brace **}** in the file.
- 2. Type a natural language comment: `// Add a method to add two numbers` and **Enter**.
- 3. GitHub Copilot generates a code suggestion for you. The suggested
- implementation shows in gray text.
- 4. To accept the suggestion, select **Tab**.
- Let's use Copilot Chat to submit a coding-related question as a prompt:
- 1. Select the **GitHub Copilot** badge in the upper-right corner of the IDE.
- 2. Select **Open Chat Window** from the dropdown.
- 3. Enter the following prompt in the chat window:
- Copilot prompt
- `Generate sample code for a simple C# method to add two numbers.`
- 4. Copilot Chat generates sample code in response to your prompt.
- GitHub Copilot is powered by AI, so surprises and mistakes are possible. For more
- information, see GitHub Copilot FAQs .
- Get started with GitHub Copilot in Visual Studio. Note that it requires Visual Studio 2022
- version 17.8 or later.
- **Use code snippets**Visual Studio provides useful *code snippets* that you can use to quickly and easily
- generate commonly used code blocks. Code snippets are available for different
- programming languages including C#, Visual Basic, and C++.
- Let's add the C# `void Main` snippet to our file.
- 1. Place your cursor just above the final closing brace **}** in the file, and type the
- characters `svm` .
- A pop-up dialog box appears with information about the `svm` code snippet.
- 2. Select the **Tab** key twice to insert the code snippet.
- You'll see the `static void Main()` method signature get added to the file. The
- Main() method is the entry point for C# applications.

- Available code snippets vary for different programming languages. You can look at the
- available code snippets for your language by choosing **Edit** > **IntelliSense** > **Insert**
- **Snippet** or by selecting the **Ctrl**+**K**, **Ctrl**+**X** keys, and then choosing the folder for your
- programming language. For C#, the snippet list looks like this:The list includes snippets for creating a class, a constructor, a for loop, an if or switch
- statement, and more.
- The Text Editor toolbar, which is the row of buttons under the menu bar in Visual Studio,
- helps make you more productive as you code. For example, you can toggle IntelliSense
- completion mode, increase or decrease a line indent, or comment out code that you
- don't want to compile.
- Let's comment out some code.
- 1. Paste the following code into the `Main()` method body.
- C#
- # Comment out code
- `// someWords is a string array.`
- `string[] someWords = {`
- `"the",`
- `"quick",`
- `"brown",`
- `"fox",`
- `"jumps"`
- `};`2. We're not using the `moreWords` variable, but we might use it later so we don't want
- to delete it. Instead, we'll comment out those lines. Select the entire definition of
- `moreWords` down to the closing semicolon, and then choose the **Comment out the**
- **selected lines** button on the Text Editor toolbar. If you prefer to use the keyboard,
- select **Ctrl**+**K**, **Ctrl**+**C**.
- The C# comment characters `//` are added to the beginning of each selected line
- to comment out the code.
- When you want to uncomment lines, you can select them, and then choose the
- **Uncomment the selected lines** button on the Text Editor toolbar. If you prefer to
- use the keyboard, select **Ctrl**+**K**, **Ctrl**+**U**.

- We don't want to see the empty constructor that was generated for `Class1` , so to
- unclutter our view of the code, let's collapse it. Choose the small gray box with the
- minus sign inside it in the margin of the first line of the constructor. Or, if you prefer to
- use the keyboard, place the cursor anywhere in the constructor code and select the
- **Ctrl**+**M**, **Ctrl**+**M** keys.
- `string[] moreWords = {`
- `"over",`
- `"the",`
- `"lazy",`
- `"dog"`
- `};`
- `// Alphabetically sort the words.`
- `IEnumerable<string> query = from word in someWords`
- `orderby word`
- `select word;`
- # Collapse code blocksThe code block collapses to just the first line, followed by an ellipsis (`...`). To expand
- the code block again, select the same gray box that now has a plus sign in it, or select
- **Ctrl**+**M**, **Ctrl**+**M** again. This feature is called Outlining and is especially useful when
- you're collapsing long methods or entire classes.
- # View symbol definitions
- The Visual Studio editor makes it easy to inspect the definition of a type, method, or
- variable. One way is to go to the definition, in whichever file has it, by choosing **Go to**
- **Definition** or by selecting the **F12** key anywhere a symbol is referenced. An even quicker
- way that doesn't move your focus away from the code you're working on is to use Peek
- Definition.
- Let's peek at the definition of the `string` type.
- 1. Right-click on any occurrence of `string` and choose **Peek Definition** from the
- content menu. Or, select the **Alt**+**F12** keys.
- A pop-up window appears with the definition of the `String` class. You can scroll
- within the pop-up window, or even peek at the definition of another type from the
- peeked code.

- 2. Close the peek definition window by choosing the small box with an "x" at the top
- right of the pop-up window.

Use IntelliSense to complete words

- IntelliSense is an invaluable resource when you're coding. It can show you information
- about available members of a type, or parameter details for different overloads of amethod. You can also use IntelliSense to complete a word after you type enough
- characters to disambiguate it.
- Let's add a line of code to print out the ordered strings to the console window, which is
- the standard place for output from the program to go.
- 1. Below the `query` variable, start typing the following code:
- C#
- You'll see an IntelliSense pop-up appear with information about the `query` symbol.
- 2. To insert the rest of the word `query` by using IntelliSense word completion, select
- the **Tab** key.
- 3. Finish off the code block to look like the following code. You can practice further
- with code snippets by entering `cw` and then selecting **Tab** twice to generate the
- `Console.WriteLine` statement.
- C#
- Nobody gets code right the first time, and one of the things you might have to change
- is the name of a variable or method. Let's try out Visual Studio's refactor functionality to
- rename the `someWords` variable to `unsortedWords` .
- `foreach (string str in qu`
- `foreach (string str in query)`
- `{`
- `Console.WriteLine(str);`
- `}`

Refactor a name

- 1. Place your cursor over the definition of the `someWords` variable, and choose
- **Rename** from the right-click or context menu, or select the **F2** key.
- A **Rename** dialog box appears at the top right of the editor.
- 2. Enter the desired name **unsortedWords**. You'll see that the reference to
- `unsortedWords` in the `query` assignment statement is also automatically renamed.

- Before you select the **Enter** key, select the **Include comments** checkbox in the
- **Rename** pop-up box.
- 3. Select the **Enter** key.
- Both occurrences of `someWords` in your code have been renamed, as well as the
- text `someWords` in your code comment.

Next steps

Feedback

- **Was this page helpful?**
- Provide product feedback
- | Ask the community
- GitHub Copilot Completions in Visual Studio
- GitHub Copilot Chat in Visual Studio
- Code snippets
- Navigate code
- Outlining
- Go To Definition and Peek Definition
- Refactoring
- Use IntelliSense

See also

- **Yes**
- **No**

Compile and build in Visual Studio

- Article • 02/03/2025
- For a first introduction to building within the IDE, see Walkthrough: Building an
- application.
- You can use any of the following methods to build an application: the Visual Studio IDE,
- the MSBuild command-line tools, and Azure Pipelines:
- **Build Method**
- **Benefits**
- IDE
- - Create builds immediately and test them in a debugger.
- - Run multi-processor builds for C++ and C# projects.
- - Customize different aspects of the build system.
- CMake
- - Build C++ projects using the CMake tool
- - Use the same build system across Linux and Windows platforms.
- MSBuild command
- line
- - Build projects without installing Visual Studio.

- - Run multi-processor builds for all project types.
- - Customize most areas of the build system.
- Azure Pipelines
- - Automate your build process as part of a continuous
- integration/continuous delivery pipeline.
- - Apply automated tests with every build.
- - Employ virtually unlimited cloud-based resources for build processes.
- - Modify the build workflow and create build activities to perform deeply
- customized tasks.
- The documentation in this section goes into further details of the IDE-based build
- process. For more information on the other methods, see CMake, MSBuild and Azure
- Pipelines, respectively.
- When you create a project, Visual Studio created default build configurations for the
- project and the solution that contains the project. These configurations define how the
- solutions and projects are built and deployed. Project configurations in particular are
- unique for a target platform (such as Windows or Linux) and build type (such as debug
- or release). You can edit these configurations however you like, and can also create your
- own configurations as needed.
- **Expand table**
- # Building from the IDEFeedback
- **Was this page helpful?**
- Provide product feedback
- | Ask the community
- For a first introduction to building within the IDE, see Walkthrough: Building an
- application.
- Next, see Building and cleaning projects and solutions in Visual Studio to learn about
- the different customizations you can make to the process. Customizations include
- changing output directories, specifying custom build events, managing project
- dependencies, managing build log files, and suppressing compiler warnings.
- From there, you can explore a variety of other tasks:
- Understand build configurations
- Configure projects to target platforms
- Manage project and solution properties.
- Specify build events in C# and Visual Basic
- Set build options

- Build multiple projects in parallel
- Building (compiling) website projects
- CMake projects in Visual Studio

Related content

- Yes
- No

Quickstart: Debug with C++ using the Visual Studio debugger

Article • 01/12/2024

The Visual Studio debugger provides many powerful features to help you debug your apps. This topic provides a quick way to learn some of the basic features.

1. Open Visual Studio and create a project.

Press **Esc** to close the start window. Type **Ctrl + Q** to open the search box, type **c++**, choose **Templates**, then choose **Create new Console App project**. In the dialog box that appears, choose **Create**.

If you don't see the **Windows Console Application** project template, go to **Tools** > **Get Tools and Features...**, which opens the Visual Studio Installer. The Visual Studio Installer launches. Choose the **Desktop development with C++** workload, then choose **Modify**.

Visual Studio creates the project.

2. In MyDbgApp.cpp, replace the following code

C++

with this code (do not remove `#include "stdafx.h"`):

C++

Create a new project

```
int main()
{
return 0;
}
#include <list>
#include <iostream>
using namespace std;
void doWork()
{
list <int> c1;
```

A *breakpoint* is a marker that indicates where Visual Studio should suspend your running

- code so you can take a look at the values of variables, or the behavior of memory, or
- whether or not a branch of code is getting run. It is the most basic feature in debugging.
- 1. To set the breakpoint, click in the gutter to the left of the `doWork` function call (or
- select the line of code and press **F9**).
- 2. Now press **F5** (or choose **Debug > Start Debugging**).
- The debugger pauses where you set the breakpoint. The statement where the
- debugger and app execution is paused is indicated by the yellow arrow. The line
- with the `doWork` function call has not yet executed.
- `c1.push_back(10);`
- `c1.push_back(20);`
- `const list <int> c2 = c1;`
- `const int &i = c2.front();`
- `const int &j = c2.front();`
- `cout << "The first element is " << i << endl;`
- `cout << "The second element is " << j << endl;`
- `}`
- `int main()`
- `{`
- `doWork();`
- `}`
- # Set a breakpoint
- **Tip**If you have a breakpoint in a loop or recursion, or if you have many
- breakpoints that you frequently step through, use a **conditional breakpoint** to
- make sure that your code is suspended ONLY when specific conditions are
- met. A conditional breakpoint saves time and can also make it easier to debug
- issues that are hard to reproduce.
- When trying to debug memory-related failures in C++, you can also use
- breakpoints to inspect address values (look for NULL) and reference counts.
- # Navigate code
- There are different commands to instruct the debugger to continue. We show a useful
- code navigation command that is available starting in Visual Studio 2017.
- While paused at the breakpoint, hover over the statement `c1.push_back(20)` until the

- green **Run to click** button appears, and then press the **Run to click** button.
- The app continues execution, calling `doWork` , and pauses on the line of code where you
- clicked the button.
- Common keyboard commands used to step through code include **F10** and **F11**. For more
- in-depth instructions, see First look at the debugger.

Inspect variables in a datatip

- 1. In the current line of code (marked by the yellow execution pointer), hover over
- the `c1` object with your mouse to show a datatip.The datatip shows you the current value of the `c1` variable and allows you to
- inspect its properties. When debugging, if you see a value you don't expect, you
- probably have a bug in the preceding or calling lines of code.
- 2. Expand the datatip to look at the current property values of the `c1` object.
- 3. If you want to pin the datatip so that you can continue to see the value of `c1` while
- you execute code, click the small pin icon. (You can move the pinned datatip to a
- convenient location.)

Edit code and continue debugging

- If you identify a change that you want to test in your code while in the middle of a
- debugging session, you can do that, too.
- 1. Click the second instance of `c2.front()` and change `c2.front()` to `c2.back()` .
- 2. Press **F10** (or **Debug > Step Over**) a few times to advance the debugger and
- execute the edited code.
- **F10** advances the debugger one statement at a time, but steps over functions
- instead of stepping into them (the code that you skip still executes).
- For more information on using edit-and-continue and on feature limitations, see Edit
- and Continue.

Feedback

- **Was this page helpful?**
- In this tutorial, you've learned how to start the debugger, step through code, and
- inspect variables. You may want to get a high-level look at debugger features along with
- links to more information.

- # Next steps
-
- Yes
- No# Write unit tests for C/C++ in Visual
- # Studio
- Article • 12/16/2024
- You can write and run your C++ unit tests by using the **Test Explorer** window. It works
- just like it does for other languages. For more information about using **Test Explorer**,
- see Run unit tests with Test Explorer.
- **Note**
- Some features such as Live Unit Testing, Coded UI Tests and IntelliTest aren't
- supported for C++.
- Visual Studio includes these C++ test frameworks with no extra downloads required:
- Microsoft Unit Testing Framework for C++
- Google Test
- Boost.Test
- CTest
- You can use the installed frameworks, or write your own test adapter for whatever
- framework you want to use within Visual Studio. A test adapter integrates unit tests with
- the **Test Explorer** window. Several non-Microsoft adapters are available on the Visual
- Studio Marketplace . For more information, see Install unit test frameworks.
- Visual Studio 2017 and later (Professional and Enterprise)
- C++ unit test projects support CodeLens.
- Visual Studio 2017 and later (all editions)
- **Google Test Adapter** is included as a default component of the **Desktop**
- **development with C++** workload. It has a project template that you can add to
- a solution. Right-click on the solution node in **Solution Explorer** and choose
- **Add** > **New Project** on the shortcut menu to add the project template. It also

- has options you can configure by using **Tools** > **Options**. For more information,
- see How to: Use Google Test in Visual Studio.
- **Boost.Test** is included as a default component of the **Desktop development**
- **with C++** workload. It's integrated with **Test Explorer**, but currently doesn'thave a project template. You must manually configure it. For more information,
- see How to: Use Boost.Test in Visual Studio.
- **CTest** support is included with the **C++ CMake tools** component, which is part
- of the **Desktop development with C++** workload. For more information, see
- How to: Use CTest in Visual Studio.
- Earlier versions of Visual Studio
- You can download the Google Test adapter and Boost.Test Adapter extensions on
- the Visual Studio Marketplace. Find them at Test adapter for Boost.Test and Test
- adapter for Google Test .
- **Tip**
- You can also use Copilot `/tests` slash command to generate unit tests from code.
- For example, you can type `/tests using Boost framework` to generate Boost.Test
- tests. For more information, see **Use slash commands in Copilot Chat**.

Basic test workflow

- The following sections show the basic steps to get you started with C++ unit testing.
- The basic configuration is similar for both the Microsoft and Google Test frameworks.
- Boost.Test requires that you manually create a test project.

Create a test project in Visual Studio 2022

- Define and run unit tests inside one or more **test projects**. A test project creates a
- separate app that calls the code in your executable and reports on its behavior. Create
- test projects in the same solution as the code you want to test.
- To add a new test project to an existing solution:
- 1. Right-click on the Solution node in **Solution Explorer**.
- 2. In the context menu, choose **Add** > **New Project**.

- 3. Set **Language** to **C++** and type *test* in the search box. The following screenshot
- shows the test projects that are available when the **Desktop Development with**
- **C++** and the **UWP Development** workload are installed:## Create references to other projects in the solution
- To enable access to the functions in the project under test, add a reference to the
- project in your test project. In **Solution Explorer**, expand your test project. Right-click
- **References** and then select **Add** > **Reference**. In the **Add Reference** dialog box, choose
- the projects you want to test.## Link to object or library files
- If the test code doesn't export the functions that you want to test, add the output `.obj`
- or `.lib` files to the dependencies of the test project. For more information, see To link
- the tests to the object or library files. Don't include object files that have a `main` function
- or another standard entry point such as `wmain` , `WinMain` , or `DllMain` . When you add new
- source files to your project, update the test project dependencies to include the
- corresponding object files.
- ## Add #include directives for header files
- In your unit test `.cpp` file, add an `#include` directive for any header files that declare the
- types and functions you want to test. Type `#include "` , and then IntelliSense activates to
- help you choose. Repeat for any more headers. **Tip**
- To avoid having to type the full path in each include statement in the source file,
- add the required folders in **Project** > **Properties** > **C/C++** > **General** > **Additional**
- **Include Directories**.
- ## Write test methods
- **Note**
- This section shows syntax for the Microsoft Unit Testing Framework for C/C++. For

- more information, see **Microsoft.VisualStudio.TestTools.CppUnitTestFramework**
- **API reference**.
- For Google Test documentation, see **Google Test primer** . For Boost.Test, see
- **Boost Test library: The unit test framework** .
- The `.cpp` file in your test project has a stub class and method defined for you. They
- show an example of how to write test code. The signatures use the TEST_CLASS and
- TEST_METHOD macros, which make the methods discoverable from the **Test Explorer**
- window.TEST_CLASS and TEST_METHOD are part of the Microsoft Native Test Framework. **Test**
- **Explorer** discovers test methods in other supported frameworks in a similar way.
- A TEST_METHOD returns void. To produce a test result, use the static methods in the
- `Assert` class to test actual results against expected results. In the following example,
- assume `MyClass` has a constructor that takes a `std::string` . This example shows how
- you can test that the constructor initializes the class the way you expect:
- C++
- In the previous example, the result of the `Assert::AreEqual` call determines whether the
- test passes or fails. The `Assert` class contains many other methods to compare expected
- results with actual results.
- You can add *traits* to test methods to specify test owners, priority, and other
- information. You can then use these values to sort and group tests in **Test Explorer**. For
- more information, see Run unit tests with Test Explorer.
- 1. On the **Test** menu, choose **Test Explorer**. The following illustration shows a test
- project before you run tests.
- `TEST_METHOD(TestClassInit)`
- `{`
- `std::string name = "Bill";`
- `MyClass mc(name);`
- `Assert::AreEqual(name, mc.GetName());`
- `}`
- # Run the tests **Note**

- CTest integration with **Test Explorer** is not yet available. Run CTest tests from
- the CMake main menu.
- 2. If any of your tests are missing from the window, build the test project by right
- clicking its node in **Solution Explorer** and choosing **Build** or **Rebuild**.
- 3. In **Test Explorer**, choose **Run All**, or select the specific tests you want to run. Right
- click on a test for other options, including running it in debug mode with
- breakpoints enabled. After all the tests run, the window shows the tests that
- passed and the ones that failed.
- For failed tests, the message displays details that help to diagnose the cause. Right-click
- on the failing test for a pop-up menu. Choose **Debug** to step through the function
- where the failure occurred.For more information on using **Test Explorer**, see Run unit tests with Test Explorer.
- For more information on unit testing, see Unit test basics.

Use CodeLens

- **Visual Studio 2017 and later (Professional and Enterprise editions)**
- CodeLens lets you quickly see the status of a unit test without leaving the code editor.
- Initialize CodeLens for a C++ unit test project in any of the following ways:
- Edit and build your test project or solution.
- Rebuild your project or solution.
- Run tests from the **Test Explorer** window.
- After you initialize CodeLens, you can see the test status icons above each unit test.
-
- Choose the icon for more information, or to run or debug the unit test:

Feedback

- **Was this page helpful?**
- Provide product feedback
- | Ask the community
- Unit test your code

Related content

- **Yes**

No Walkthrough: Compiling a Native C++ Program on the Command Line

Article • 02/08/2022

Visual Studio includes a command-line C and C++ compiler. You can use it to create everything from basic console apps to Universal Windows Platform apps, Desktop apps, device drivers, and .NET components. In this walkthrough, you create a basic, "Hello, World"-style C++ program by using a text editor, and then compile it on the command line. If you'd like to try the Visual Studio IDE instead of using the command line, see Walkthrough: Working with Projects and Solutions (C++) or Using the Visual Studio IDE for C++ Desktop Development. In this walkthrough, you can use your own C++ program instead of typing the one that's shown. Or, you can use a C++ code sample from another help article.

Prerequisites

To complete this walkthrough, you must have installed either Visual Studio and the optional **Desktop development with C++** workload, or the command-line Build Tools for Visual Studio. Visual Studio is an *integrated development environment* (IDE). It supports a full-featured editor, resource managers, debuggers, and compilers for many languages and platforms. Versions available include the free Visual Studio Community edition, and all can support C and C++ development. For information on how to download and install Visual Studio, see Install C++ support in Visual Studio. The Build Tools for Visual Studio installs only the command-line compilers, tools, and libraries you need to build C and C++ programs. It's perfect for build labs or classroom

- exercises and installs relatively quickly. To install only the command-line tools, look for
- Build Tools for Visual Studio on the Visual Studio Downloads page.
- Before you can build a C or C++ program on the command line, verify that the tools are
- installed, and you can access them from the command line. Visual C++ has complex
- requirements for the command-line environment to find the tools, headers, and libraries
- it uses. **You can't use Visual C++ in a plain command prompt window** without doing
- some preparation. Fortunately, Visual C++ installs shortcuts for you to launch a
- developer command prompt that has the environment set up for command line builds.
- Unfortunately, the names of the developer command prompt shortcuts and wherethey're located are different in almost every version of Visual C++ and on different
- versions of Windows. Your first walkthrough task is finding the right one to use.
- **Note**
- A developer command prompt shortcut automatically sets the correct paths for the
- compiler and tools, and for any required headers and libraries. You must set these
- environment values yourself if you use a regular **Command Prompt** window. For
- more information, see **Use the MSVC toolset from the command line**. We
- recommend you use a developer command prompt shortcut instead of building
- your own.

Open a developer command prompt

- 1. If you have installed Visual Studio 2017 or later on Windows 10 or later, open the
- Start menu and choose **All apps**. Scroll down and open the **Visual Studio** folder
- (not the Visual Studio application). Choose **Developer Command Prompt for VS** to
- open the command prompt window.
- If you have installed Microsoft Visual C++ Build Tools 2015 on Windows 10 or
- later, open the **Start** menu and choose **All apps**. Scroll down and open the **Visual**

- **C++ Build Tools** folder. Choose **Visual C++ 2015 x86 Native Tools Command**
- **Prompt** to open the command prompt window.
- You can also use the Windows search function to search for "developer command
- prompt" and choose one that matches your installed version of Visual Studio. Use
- the shortcut to open the command prompt window.
- 2. Next, verify that the Visual C++ developer command prompt is set up correctly. In
- the command prompt window, enter `cl` and verify that the output looks
- something like this:
- Output
- `C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise>cl`
- `Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x86`
- `Copyright (C) Microsoft Corporation. All rights reserved.`
- `usage: cl [ option... ] filename... [ /link linkoption... ]`
- There may be differences in the current directory or version numbers. These values
- depend on the version of Visual C++ and any updates installed. If the aboveoutput is similar to what you see, then you're ready to build C or C++ programs at
- the command line.
- **Note**
- If you get an error such as "'cl' is not recognized as an internal or external
- command, operable program or batch file," error C1034, or error LNK1104
- when you run the **cl** command, then either you are not using a developer
- command prompt, or something is wrong with your installation of Visual C++.
- You must fix this issue before you can continue.
- If you can't find the developer command prompt shortcut, or if you get an error
- message when you enter `cl` , then your Visual C++ installation may have a
- problem. Try reinstalling the Visual C++ component in Visual Studio, or reinstall
- the Microsoft Visual C++ Build Tools. Don't go on to the next section until the **cl**
- command works. For more information about installing and troubleshooting Visual
- C++, see Install Visual Studio.
- **Note**
- Depending on the version of Windows on the computer and the system

- security configuration, you might have to right-click to open the shortcut
- menu for the developer command prompt shortcut and then choose **Run as**
- **administrator** to successfully build and run the program that you create by
- following this walkthrough.
- # Create a Visual C++ source file and compile it on the
- # command line
- 1. In the developer command prompt window, enter `md c:\hello` to create a
- directory, and then enter `cd c:\hello` to change to that directory. This directory is
- where both your source file and the compiled program get created.
- 2. Enter `notepad hello.cpp` in the command prompt window.
- Choose **Yes** when Notepad prompts you to create a new file. This step opens a
- blank Notepad window, ready for you to enter your code in a file named hello.cpp.
- 3. In Notepad, enter the following lines of code:C++
- This code is a simple program that will write one line of text on the screen and
- then exit. To minimize errors, copy this code and paste it into Notepad.
- 4. Save your work! In Notepad, on the **File** menu, choose **Save**.
- Congratulations, you've created a C++ source file, hello.cpp, that is ready to
- compile.
- 5. Switch back to the developer command prompt window. Enter `dir` at the
- command prompt to list the contents of the c:\hello directory. You should see the
- source file hello.cpp in the directory listing, which looks something like:
- Output
- The dates and other details will differ on your computer.
- `#include <iostream>`
- `using namespace std;`
- `int main()`
- `{`
- `cout << "Hello, world, from Visual C++!" << endl;`
- `}`
- `c:\hello>dir`
- `Volume in drive C has no label.`
- `Volume Serial Number is CC62-6545`
- `Directory of c:\hello`
- `05/24/2016 05:36 PM <DIR> .`

- `05/24/2016 05:36 PM <DIR> ..`
- `05/24/2016 05:37 PM 115 hello.cpp`
- `1 File(s) 115 bytes`
- `2 Dir(s) 571,343,446,016 bytes free`
- **Note**
- If you don't see your source code file, *hello.cpp* , make sure the current
- working directory in your command prompt is the *C:\hello* directory you
- created. Also make sure that this is the directory where you saved your source
- file. And make sure that you saved the source code with a *.cpp* file name
- extension, not a *.txt* extension. Your source file gets saved in the current
- directory as a *.cpp* file automatically if you open Notepad at the commandprompt by using the **notepad hello.cpp** command. Notepad's behavior is
- different if you open it another way: By default, Notepad appends a *.txt*
- extension to new files when you save them. It also defaults to saving files in
- your *Documents* directory. To save your file with a *.cpp* extension in Notepad,
- choose **File** > **Save As**. In the **Save As** dialog, navigate to your *C:\hello* folder
- in the directory tree view control. Then use the **Save as type** dropdown
- control to select **All Files (*.*)**. Enter *hello.cpp* in the **File name** edit control,
- and then choose **Save** to save the file.
- 6. At the developer command prompt, enter `cl /EHsc hello.cpp` to compile your
- program.
- The cl.exe compiler generates an .obj file that contains the compiled code, and
- then runs the linker to create an executable program named hello.exe. This name
- appears in the lines of output information that the compiler displays. The output of
- the compiler should look something like:
- Output
- `c:\hello>cl /EHsc hello.cpp`
- `Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x86`
- `Copyright (C) Microsoft Corporation. All rights reserved.`
- `hello.cpp`
- `Microsoft (R) Incremental Linker Version 14.10.25017.0`
- `Copyright (C) Microsoft Corporation. All rights reserved.`
- `/out:hello.exe`
- `hello.obj`

- **Note**
- If you get an error such as "'cl' is not recognized as an internal or external
- command, operable program or batch file," error C1034, or error LNK1104,
- your developer command prompt is not set up correctly. For information on
- how to fix this issue, go back to the **Open a developer command prompt**
- section.
- **Note**
- If you get a different compiler or linker error or warning, review your source
- code to correct any errors, then save it and run the compiler again. Forinformation about specific errors, use the search box to look for the error
- number.
- 7. To run the hello.exe program, at the command prompt, enter `hello` .
- The program displays this text and exits:
- Output
- `Hello, world, from Visual C++!`
- Congratulations, you've compiled and run a C++ program by using the command
- line tools.

Next steps

- This "Hello, World" example is about as simple as a C++ program can get. Real world
- programs usually have header files, more source files, and link to libraries.
- You can use the steps in this walkthrough to build your own C++ code instead of typing
- the sample code shown. These steps also let you build many C++ code sample
- programs that you find elsewhere. You can put your source code and build your apps in
- any writeable directory. By default, the Visual Studio IDE creates projects in your user
- folder, in a *source**repos* subfolder. Older versions may put projects in a
- *Documents**Visual Studio <version>**Projects* folder.
- To compile a program that has additional source code files, enter them all on the
- command line, like:
- `cl /EHsc file1.cpp file2.cpp file3.cpp`
- The `/EHsc` command-line option instructs the compiler to enable standard C++

- exception handling behavior. Without it, thrown exceptions can result in undestroyed
- objects and resource leaks. For more information, see /EH (Exception Handling Model).
- When you supply additional source files, the compiler uses the first input file to create
- the program name. In this case, it outputs a program called file1.exe. To change the
- name to program1.exe, add an /out linker option:
- `cl /EHsc file1.cpp file2.cpp file3.cpp /link /out:program1.exe`
- And to catch more programming mistakes automatically, we recommend you compile
- by using either the /W3 or /W4 warning level option:`cl /W4 /EHsc file1.cpp file2.cpp file3.cpp /link /out:program1.exe`
- The compiler, cl.exe, has many more options. You can apply them to build, optimize,
- debug, and analyze your code. For a quick list, enter `cl /?` at the developer command
- prompt. You can also compile and link separately and apply linker options in more
- complex build scenarios. For more information on compiler and linker options and
- usage, see C/C++ Building Reference.
- You can use NMAKE and makefiles, MSBuild and project files, or CMake, to configure
- and build more complex projects on the command line. For more information on using
- these tools, see NMAKE Reference, MSBuild, and CMake projects in Visual Studio.
- The C and C++ languages are similar, but not the same. The MSVC compiler uses a
- simple rule to determine which language to use when it compiles your code. By default,
- the MSVC compiler treats files that end in *.c* as C source code, and files that end in
- *.cpp* as C++ source code. To force the compiler to treat all files as C++ independent of
- file name extension, use the /TP compiler option.
- The MSVC compiler includes a C Runtime Library (CRT) that conforms to the ISO C99
- standard, with minor exceptions. Portable code generally compiles and runs as expected.
- Certain obsolete library functions, and several POSIX function names, are deprecated by
- the MSVC compiler. The functions are supported, but the preferred names have

- changed. For more information, see Security Features in the CRT and Compiler Warning
- (level 3) C4996.

See also

- C++ Language Reference
- Projects and build systems
- MSVC Compiler Options

Walkthrough: Compile a C program on the command line

Article • 05/10/2022

The Visual Studio build tools include a C compiler that you can use to create everything from basic console programs to full Windows Desktop applications, mobile apps, and more. Microsoft C/C++ (MSVC) is a C and C++ compiler that, in its latest versions, conforms to some of the latest C language standards, including C11 and C17.

This walkthrough shows how to create a basic, "Hello, World"-style C program by using a text editor, and then compile it on the command line. If you'd rather work in C++ on the command line, see Walkthrough: Compiling a Native C++ Program on the Command Line. If you'd like to try the Visual Studio IDE instead of using the command line, see Walkthrough: Working with Projects and Solutions (C++) or Using the Visual Studio IDE for C++ Desktop Development.

Prerequisites

To complete this walkthrough, you must have installed either Visual Studio or the Build Tools for Visual Studio and the optional Desktop development with C++ workload. Visual Studio is a powerful integrated development environment that supports a full featured editor, resource managers, debuggers, and compilers for many languages and platforms. For information on these features and how to download and install Visual

- Studio, including the free Visual Studio Community edition, see Install Visual Studio.
- The Build Tools for Visual Studio version of Visual Studio installs only the command-line
- toolset, the compilers, tools, and libraries you need to build C and C++ programs. It's
- perfect for build labs or classroom exercises and installs relatively quickly. To install only
- the command-line toolset, download Build Tools for Visual Studio from the Visual Studio
- downloads page and run the installer. In the Visual Studio installer, select the **Desktop**
- **development with C++** workload (in older versions of Visual Studio, select the **C++**
- **build tools** workload), and choose **Install**.
- When you've installed the tools, there's another tool you'll use to build a C or C++
- program on the command line. MSVC has complex requirements for the command-line
- environment to find the tools, headers, and libraries it uses. **You can't use MSVC in a**
- **plain command prompt window** without some preparation. You need a *developer*
- *command prompt* window, which is a regular command prompt window that has all the
- required environment variables set. Fortunately, Visual Studio installs shortcuts for youto launch developer command prompts that have the environment set up for command
- line builds. Unfortunately, the names of the developer command prompt shortcuts and
- where they're located are different in almost every version of Visual Studio and on
- different versions of Windows. Your first walkthrough task is to find the right shortcut to
- use.
- **Note**
- A developer command prompt shortcut automatically sets the correct paths for the
- compiler and tools, and for any required headers and libraries. Some of these
- values are different for each build configuration. You must set these environment
- values yourself if you don't use one of the shortcuts. For more information, see **Use**
- **the MSVC toolset from the command line**. Because the build environment is

- complex, we strongly recommend you use a developer command prompt shortcut
- instead of building your own.
- These instructions vary depending on which version of Visual Studio you're using. To see
- the documentation for your preferred version of Visual Studio, use the **Version** selector
- control. It's found at the top of the table of contents on this page.

Open a developer command prompt in Visual Studio 2022

- If you've installed Visual Studio 2022 on Windows 10 or later, open the Start menu, and
- choose **All apps**. Then, scroll down and open the **Visual Studio 2022** folder (not the
- Visual Studio 2022 app). Choose **Developer Command Prompt for VS 2022** to open the
- command prompt window.
- If you're using a different version of Windows, look in your Start menu or Start page for
- a Visual Studio tools folder that contains a developer command prompt shortcut. You
- can also use the Windows search function to search for "developer command prompt"
- and choose one that matches your installed version of Visual Studio. Use the shortcut to
- open the command prompt window.
- Next, verify that the developer command prompt is set up correctly. In the command
- prompt window, enter `cl` (or `CL` , case doesn't matter for the compiler name, but it does
- matter for compiler options). The output should look something like this:
- Output
```
C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise>cl
Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x86
Copyright (C) Microsoft Corporation. All rights reserved.
usage: cl [ option... ] filename... [ /link linkoption... ]
```
- There may be differences in the current directory or version numbers, depending on the
- version of Visual Studio and any updates installed. If the above output is similar to what
- you see, then you're ready to build C or C++ programs at the command line.

- **Note**
- If you get an error such as "'cl' is not recognized as an internal or external
- command, operable program or batch file," error C1034, or error LNK1104 when
- you run the **cl** command, then either you are not using a developer command
- prompt, or something is wrong with your installation of Visual Studio. You must fix
- this issue before you can continue.
- If you can't find the developer command prompt shortcut, or if you get an error
- message when you enter `cl` , then your Visual Studio installation may have a problem. If
- you're using Visual Studio 2017 or later, try reinstalling the **Desktop development with**
- **C++** workload in the Visual Studio installer. For details, see Install C++ support in Visual
- Studio. Or, reinstall the Build Tools from the Visual Studio downloads page. Don't go
- on to the next section until the `cl` command works. For more information about
- installing and troubleshooting Visual Studio, see Install Visual Studio.
- **Note**
- Depending on the version of Windows on the computer and the system security
- configuration, you might have to right-click to open the shortcut menu for the
- developer command prompt shortcut and then choose **Run as Administrator** to
- successfully build and run the program that you create by following this
- walkthrough.
- # Create a C source file and compile it on the
- # command line
- 1. In the developer command prompt window, enter `cd c:\` to change the current
- working directory to the root of your C: drive. Next, enter `md c:\hello` to create adirectory, and then enter `cd c:\hello` to change to that directory. This directory
- will hold your source file and the compiled program.

- 2. Enter `notepad hello.c` at the developer command prompt. In the Notepad alert
- dialog that pops up, choose **Yes** to create a new *hello.c* file in your working
- directory.
- 3. In Notepad, enter the following lines of code:
- C
- 4. On the Notepad menu bar, choose **File** > **Save** to save *hello.c* in your working
- directory.
- 5. Switch back to the developer command prompt window. Enter `dir` at the
- command prompt to list the contents of the *c:\hello* directory. You should see
- the source file *hello.c* in the directory listing, which looks something like:
- Output
- The dates and other details will differ on your computer. If you don't see your
- source code file, *hello.c* , make sure you've changed to the *c:\hello* directory
- you created, and in Notepad, make sure that you saved your source file in this
- `#include <stdio.h>`
- `int main()`
- `{`
- `printf("Hello, World! This is a native C program compiled on the`
- `command line.\n");`
- `return 0;`
- `}`
- `C:\hello>dir`
- `Volume in drive C has no label.`
- `Volume Serial Number is CC62-6545`
- `Directory of C:\hello`
- `10/02/2017 03:46 PM <DIR> .`
- `10/02/2017 03:46 PM <DIR> ..`
- `10/02/2017 03:36 PM 143 hello.c`
- `1 File(s) 143 bytes`
- `2 Dir(s) 514,900,566,016 bytes free`directory. Also make sure that you saved the source code with a *.c* file name
- extension, not a *.txt* extension.
- 6. To compile your program, enter `cl hello.c` at the developer command prompt.
- You can see the executable program name, hello.exe, in the lines of output
- information that the compiler displays:
- Output
- `c:\hello>cl hello.c`
- `Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x86`

- `Copyright (C) Microsoft Corporation. All rights reserved.`
- `hello.c`
- `Microsoft (R) Incremental Linker Version 14.10.25017.0`
- `Copyright (C) Microsoft Corporation. All rights reserved.`
- `/out:hello.exe`
- `hello.obj`
- **Note**
- If you get an error such as "'cl' is not recognized as an internal or external
- command, operable program or batch file," error C1034, or error LNK1104,
- your developer command prompt is not set up correctly. For information on
- how to fix this issue, go back to the **Open a developer command prompt**
- section.
- If you get a different compiler or linker error or warning, review your source
- code to correct any errors, then save it and run the compiler again. For
- information about specific errors, use the search box at the top of this page to
- look for the error number.
- 7. To run your program, enter `hello` at the command prompt.
- The program displays this text and then exits:
- Output
- `Hello, World! This is a native C program compiled on the command line.`
- Congratulations, you've compiled and run a C program by using the command
- line.# Next steps
- This "Hello, World" example is about as basic as a C program can get. Real world
- programs have header files and more source files, link in libraries, and do useful work.
- You can use the steps in this walkthrough to build your own C code instead of typing
- the sample code shown. You can also build many C code sample programs that you find
- elsewhere. To compile a program that has more source code files, enter them all on the
- command line:
- `cl file1.c file2.c file3.c`
- The compiler outputs a program called *file1.exe* . To change the name to
- *program1.exe* , add an /out linker option:
- `cl file1.c file2.c file3.c /link /out:program1.exe`
- And to catch more programming mistakes automatically, we recommend you compile

- by using either the /W3 or /W4 warning level option:
- `cl /W4 file1.c file2.c file3.c /link /out:program1.exe`
- The compiler, cl.exe, has many more options you can apply to build, optimize, debug,
- and analyze your code. For a quick list, enter `cl /?` at the developer command prompt.
- You can also compile and link separately and apply linker options in more complex build
- scenarios. For more information on compiler and linker options and usage, see C/C++
- Building Reference.
- You can use NMAKE and makefiles, or MSBuild and project files to configure and build
- more complex projects on the command line. For more information on using these
- tools, see NMAKE Reference and MSBuild.
- The C and C++ languages are similar, but not the same. The Microsoft C/C++ compiler
- (MSVC) uses a basic rule to determine which language to use when it compiles your
- code. By default, the MSVC compiler treats all files that end in *.c* as C source code, and
- all files that end in *.cpp* as C++ source code. To force the compiler to treat all files as C
- no matter the file name extension, use the /TC compiler option.
- By default, MSVC is compatible with the ANSI C89 and ISO C99 standards, but not
- strictly conforming. In most cases, portable C code will compile and run as expected.
- The compiler provides optional support for the changes in ISO C11/C17. To compile with
- C11/C17 support, use the compiler flag **/std:c11** or **/std:c17** . C11/C17 support requires
- Windows SDK 10.0.20201.0 or later. Windows SDK 10.0.22000.0 or later isrecommended. You can download the latest SDK from the Windows SDK page. For more
- information, and instructions on how to install and use this SDK for C development, see
- Install C11 and C17 support in Visual Studio.
- Certain library functions and POSIX function names are deprecated by MSVC. The
- functions are supported, but the preferred names have changed. For more information,
- see Security Features in the CRT and Compiler Warning (level 3) C4996.

- # See also
- Walkthrough: Creating a Standard C++ Program (C++)
- C Language Reference
- Projects and build systems
- Compatibility# Walkthrough: Compiling a C++/CX
- # Program on the Command Line
- Article • 03/01/2023
- **Note**
- For new UWP apps and components, we recommend that you use **C++/WinRT**, a
- standard C++17 language projection for Windows Runtime APIs. C++/WinRT is
- available in the Windows SDK from version 1803 (10.0.17134.0) onward.
- C++/WinRT is implemented entirely in header files, and is designed to provide you
- with first-class access to the modern Windows API.
- The Microsoft C++ compiler (MSVC) supports C++ component extensions (C++/CX),
- which has additional types and operators to target the Windows Runtime programming
- model. You can use C++/CX to build apps for Universal Windows Platform (UWP), and
- Windows desktop. For more information, see A Tour of C++/CX and Component
- Extensions for Runtime Platforms.
- In this walkthrough, you use a text editor to create a basic C++/CX program, and then
- compile it on the command line. (You can use your own C++/CX program instead of
- typing the one that's shown, or you can use a C++/CX code sample from another help
- article. This technique is useful for building and testing small modules that have no UI
- elements.)
- **Note**
- You can also use the Visual Studio IDE to compile C++/CX programs. Because the

- IDE includes design, debugging, emulation, and deployment support that isn't
- available on the command line, we recommend that you use the IDE to build
- Universal Windows Platform (UWP) apps. For more information, see **Create a UWP**
- **app in C++**.
- # Prerequisites
- You understand the fundamentals of the C++ language.
- # Compiling a C++/CX ProgramTo enable compilation for C++/CX, you must use the /ZW compiler option. The MSVC
- compiler generates an .exe file that targets the Windows Runtime, and links to the
- required libraries.
- ## To compile a C++/CX application on the command line
- 1. Open a **Developer Command Prompt** window. For specific instructions, see To
- open a developer command prompt window.
- Administrator credentials may be required to successfully compile the code,
- depending on the computer's operating system and configuration. To run the
- command prompt window as an administrator, right-click to open the shortcut
- menu for the command prompt and then choose **More** > **Run as administrator**.
- 2. Change the current working directory in the command prompt window to a
- directory you can write to, such as your Documents directory.
- 3. At the command prompt, enter **notepad basiccx.cpp**.
- Choose **Yes** when you're prompted to create a file.
- 4. In Notepad, enter these lines:
- C++
- `using namespace Platform;`
- `int main(Platform::Array<Platform::String^>^ args)`
- `{`
- `Platform::Details::Console::WriteLine("This is a C++/CX program.");`
- `}`
- 5. On the menu bar, choose **File** > **Save**.
- You've created a C++ source file that uses the Windows Runtime Platform
- namespace namespace.

- 6. At the command prompt, enter `cl /EHsc /ZW basiccx.cpp /link`
- `/SUBSYSTEM:CONSOLE` . The `cl.exe` compiler compiles the source code into an `.obj`
- file, and then runs the linker to generate an executable program named
- basiccx.exe. The /EHsc compiler option specifies the C++ exception-handling
- model, and the /link flag specifies a console application.
- 7. To run the basiccx.exe program, at the command prompt, enter **basiccx**.The program displays this text and exits:
- Output
- `This is a C++/CX program.`

See also

- Projects and build systems
- MSVC Compiler Options

Walkthrough: Compiling a C++/CLI Program on the Command Line

- Article • 02/24/2023
- You can create Visual C++ programs that target the Common Language Runtime (CLR)
- and use the .NET Framework, and build them on the command line. Visual C++ supports
- the C++/CLI programming language, which has additional types and operators to target
- the .NET programming model. For general information about the C++/CLI language, see
- .NET Programming with C++/CLI (Visual C++).
- In this walkthrough, you use a text editor to create a basic C++/CLI program, and then
- compile it on the command line. (You can use your own C++/CLI program instead of
- typing the one that's shown, or you can use a C++/CLI code sample from another help
- article. This technique is useful for building and testing small modules that have no UI
- elements.)

Prerequisites

- You understand the fundamentals of the C++ language.

Compiling a C++/CLI Program

The following steps show how to compile a C++/CLI console application that uses .NET Framework classes.

To enable compilation for C++/CLI, you must use the /clr compiler option. The MSVC compiler generates an .exe file that contains MSIL code—or mixed MSIL and native code—and links to the required .NET Framework libraries.

To compile a C++/CLI application on the command line

1. Open a **Developer Command Prompt** window. For specific instructions, see To open a developer command prompt window.

Administrator credentials may be required to successfully compile the code, depending on the computer's operating system and configuration. To run the command prompt window as an administrator, right-click to open the shortcut menu for the command prompt and then choose **More** > **Run as administrator**.2. Change the current working directory in the command prompt window to a directory you can write to, such as your Documents directory.

3. At the command prompt, enter `notepad basicclr.cpp .` Choose **Yes** when you're prompted to create a file.

4. In Notepad, enter these lines:

```
C++
int main()
{
System::Console::WriteLine("This is a C++/CLI program.");
}
```

5. On the menu bar, choose **File** > **Save**.

You've created a Visual C++ source file that uses a .NET Framework class (Console) in the System namespace.

6. At the command prompt, enter `cl /clr basicclr.cpp` . The cl.exe compiler compiles the source code into an .obj file that contains MSIL, and then runs the linker to generate an executable program named basicclr.exe.

7. To run the basicclr.exe program, at the command prompt, enter `basicclr .`

The program displays this text and exits:

```
Output
```

- `This is a C++/CLI program.`
- ## See also
- C++ Language Reference
- Projects and build systems
- MSVC Compiler Options

2.

-

C++ Standard Library reference (STL)

- Article
- 08/17/2022
-

A C++ program can call on a large number of functions from this conforming implementation of the C++ Standard Library. These functions perform services such as input and output and provide efficient implementations of frequently used operations.

For more information about linking with the appropriate Visual C++ runtime `.lib` file, see C runtime (CRT) and C++ Standard Library (STL) `.lib` files.

Note

Microsoft's implementation of the C++ Standard Library is often referred to as the *STL* or *Standard Template Library*. Although *C++ Standard Library* is the official name of the library as defined in ISO 14882, due to the popular use of "STL" and "Standard Template Library" in search engines, we occasionally use those names to make it easier to find our documentation.

From a historical perspective, "STL" originally referred to the Standard Template Library written by Alexander Stepanov. Parts of that library were standardized in the C++ Standard Library, along with the ISO C runtime library, parts of the Boost library, and other functionality. Sometimes "STL" is used to refer to the containers and algorithms parts of the C++ Standard

Library adapted from Stepanov's STL. In this documentation, Standard Template Library (STL) refers to the C++ Standard Library as a whole.

In this section

C++ Standard Library overview Provides an overview of the Microsoft implementation of the C++ Standard Library.

`iostream` programming Provides an overview of `iostream` programming.

Header files reference Provides links to reference topics about the C++ Standard Library header files, with code examples.

Use the Microsoft C++ toolset from the command line

- Article
- 03/02/2023
-

In this article

1. Download and install the tools
2. How to use the command-line tools
3. Path and environment variables for command-line builds
4. Developer command prompt shortcuts

You can build C and C++ applications on the command line by using tools that are included in Visual Studio. The Microsoft C++ (MSVC) compiler toolset is also downloadable as a standalone package. You don't need to install the Visual Studio IDE if you don't plan to use it.

Note

This article is about how to set up an environment to use the individual compilers, linkers, librarian, and other basic tools. The native project build system in Visual Studio, based on MSBuild, doesn't use the environment as described in this article. For more information on how to use MSBuild from the command line, see MSBuild on the command line - C++.

Download and install the tools

If you've installed Visual Studio and a C++ workload, you have all the command-line tools. For information on how to install C++ and Visual Studio, see Install C++ support in Visual Studio. If you only want the command-line toolset, download the Build Tools for Visual Studio. When you run the downloaded executable, it updates and runs the Visual Studio Installer. To install only the tools you need for C++ development, select the **Desktop development with C++** workload. You can select optional libraries and toolsets to include under **Installation details**. To build code by using the Visual Studio 2015, 2017, or 2019 toolsets, select the optional MSVC v140, v141, or v142 build tools. When you're satisfied with your selections, choose **Install**.

How to use the command-line tools

When you choose one of the C++ workloads in the Visual Studio Installer, it installs the Visual Studio *platform toolset*. A platform toolset has all the C and C++ tools for a specific Visual Studio version. The tools include the C/C++ compilers, linkers, assemblers, and other build tools, and matching libraries and header files. You can use all of these tools at the command line. They're also used internally by the Visual Studio IDE. There are separate x86-hosted and x64-hosted compilers and tools to build code for x86, x64, ARM, and ARM64 targets. Each set of tools for a particular host and target build architecture is stored in its own directory.

To work correctly, the tools require several specific environment variables to be set. These variables are used to add the tools to the path, and to set the locations of include files, library files, and SDKs. To make it easy to set these environment variables, the installer creates customized *command files*, or batch files, during installation. You can run one of these command files to set a specific host and target build architecture, Windows SDK version, and platform toolset. For convenience, the installer also creates shortcuts in your Start menu. The shortcuts open developer command prompt windows by using these command files for specific combinations of host and target. These shortcuts ensure all the required environment variables are set and ready to use.

The required environment variables are specific to your installation and to the build architecture you choose. They also might be changed by product updates or upgrades. This variability is one reason why we recommend you use an installed command prompt shortcut or command file, instead of setting the environment variables yourself.

The toolsets, command files, and shortcuts installed depend on your computer processor and the options you selected during installation. The x86-hosted tools and cross tools that build x86 and x64 code are always

installed. If you have 64-bit Windows, the x64-hosted tools and cross tools that build x86 and x64 code are also installed. If you choose the optional C++ Universal Windows Platform tools, then the x86 and x64 tools that build ARM and ARM64 code also get installed. Other workloads may install these and other tools.

Path and environment variables for command-line builds

The MSVC command-line tools use the `PATH`, `TMP`, `INCLUDE`, `LIB`, and `LIBPATH` environment variables, and also use other environment variables specific to your installed tools, platforms, and SDKs. Even a simple Visual Studio installation may set twenty or more environment variables. This complexity is why we strongly recommend that you use a developer command prompt shortcut or one of the customized command files. We don't recommend you set these variables in the Windows environment yourself.

To see which environment variables are set by a developer command prompt shortcut, you can use the `SET` command. Open a plain command prompt window and capture the output of the `SET` command for a baseline. Open a developer command prompt window and capture the output of the `SET` command for comparison. Use a diff tool such as the one built into Visual Studio to highlight the environment variables set by the developer command prompt. For more information about the compiler and linker environment variables, see CL environment variables.

Developer command prompt shortcuts

The command prompt shortcuts are installed in a version-specific Visual Studio folder in your Windows Start menu. Here's a list of the base command prompt shortcuts and the build architectures they support:

- **Developer Command Prompt** - Sets the environment to use 32-bit, x86-native tools to build 32-bit, x86-native code.
- **x86 Native Tools Command Prompt** - Sets the environment to use 32-bit, x86-native tools to build 32-bit, x86-native code.
- **x64 Native Tools Command Prompt** - Sets the environment to use 64-bit, x64-native tools to build 64-bit, x64-native code.
- **x86_x64 Cross Tools Command Prompt** - Sets the environment to use 32-bit, x86-native tools to build 64-bit, x64-native code.
- **x64_x86 Cross Tools Command Prompt** - Sets the environment to use 64-bit, x64-native tools to build 32-bit, x86-native code.

The Start menu folder and shortcut names vary depending on the installed version of Visual Studio. If you set one, they also depend on the installation

Nickname. For example, suppose you installed Visual Studio 2022, and you gave it a nickname of *Latest*. The developer command prompt shortcut is named **Developer Command Prompt for VS 2022 (Latest)**, in a folder named **Visual Studio 2022**.

Note

Several command-line tools or tool options may require Administrator permission. If you have permission issues when you use them, we recommend that you open the developer command prompt window by using the **Run as Administrator** option. Right-click to open the shortcut menu for the command prompt window, then choose **More**, **Run as administrator**.

To open a developer command prompt window

1. On the desktop, open the Windows **Start** menu. In Windows 11, choose the **All apps** button to open the list of installed apps. In Windows 10, the list is open to the left. Scroll down the list to find and open the folder (not the app) for your version of Visual Studio, for example, **Visual Studio 2022**.

2. In the folder, choose the **Developer Command Prompt** for your version of Visual Studio. This shortcut starts a developer command prompt window that uses the default build architecture of 32-bit, x86-native tools to build 32-bit, x86-native code. If you prefer a non-default build architecture, choose one of the native or cross tools command prompts to specify the host and target architecture.

For an even faster way to open a developer command prompt, enter *developer command prompt* in the desktop search box. Then choose the result you want.

Note

By default, the current working directory in a developer command prompt is the root of your Visual Studio installation in the Program Files directory. This isn't an appropriate location for your code and projects. Change the current working directory to another location before you create a project. The IDE creates projects in your user directory, typically in *%USERPROFILE%\source\repos*.

Developer command file locations

If you prefer to set the build environment in an existing command prompt window, you can use one of the command files created by the installer. We recommend you set the environment in a new command prompt window. We don't recommend you later switch environments in the same command window.

The command file location depends on the version of Visual Studio you installed, and on choices you made during installation. For Visual Studio 2019, the typical installation location on a 64-bit system is in *\Program Files\Microsoft Visual Studio\2022\<edition>*. The *<edition>* may be Community, Professional, Enterprise, BuildTools, or another nickname you supplied.

The primary developer command prompt command file, *VsDevCmd.bat*, is located in the *Common7\Tools* subdirectory. When no parameters are specified, it sets the environment to use the x86-native tools to build 32-bit x86 code.

More command files are available to set up specific build architectures. The command files available depend on the Visual Studio workloads and options you've installed. In Visual Studio 2017 and Visual Studio 2019, you'll find them in the VC\Auxiliary\Build subdirectory.

These command files set default parameters and call *VsDevCmd.bat* to set up the specified build architecture environment. A typical installation may include these command files:

Command File	Host and Target architectures
vcvars32.bat	Use the 32-bit x86-native tools to build 32-bit x86 code.
vcvars64.bat	Use the 64-bit x64-native tools to build 64-bit x64 code.
vcvarsx86_amd64.bat	Use the 32-bit x86-native cross tools to build 64-bit x64 code.
vcvarsamd64_x86.bat	Use the 64-bit x64-native cross tools to build 32-bit x86 code.
vcvarsx86_arm.bat	Use the 32-bit x86-native cross tools to build ARM code.
vcvarsamd64_arm.bat	Use the 64-bit x64-native cross tools to build ARM code.
vcvarsx86_arm64.bat	Use the 32-bit x86-native cross tools to build ARM64 code.
vcvarsamd64_arm64.bat	Use the 64-bit x64-native cross tools to build ARM64 code.
vcvarsall.bat	Use parameters to specify the host and target architectures, Windows SDK, and platform choices. For a list of supported options, call by using a `/help` parameter.

Caution

The *vcvarsall.bat* file and other Visual Studio command files can vary from computer to computer. Do not replace a missing or damaged *vcvarsall.bat* file by using a file from another computer. Rerun the Visual Studio installer to replace the missing file.

The *vcvarsall.bat* file also varies from version to version. If the current version of Visual Studio is installed on a computer that also has an earlier version of Visual Studio, do not run *vcvarsall.bat* or another Visual Studio command file from different versions in the same command prompt window.

Use the developer tools in an existing command window

The simplest way to specify a particular build architecture in an existing command window is to use the *vcvarsall.bat* file. Use *vcvarsall.bat* to set environment variables to configure the command line for native 32-bit or 64-bit compilation. Arguments let you specify cross-compilation to x86, x64, ARM, or ARM64 processors. You can target Microsoft Store, Universal Windows Platform, or Windows Desktop platforms. You can even specify which Windows SDK to use, and select the platform toolset version.

When used with no arguments, *vcvarsall.bat* configures the environment variables to use the current x86-native compiler for 32-bit Windows Desktop targets. You can add arguments to configure the environment to use any of the native or cross compiler tools. *vcvarsall.bat* displays an error message if you specify a configuration that's not installed, or not available on your computer.

`vcvarsall` syntax

vcvarsall.bat [*architecture*] [*platform_type*] [*winsdk_version*] [**-vcvars_ver=***vcversion*] [*spectre_mode*]

architecture
This optional argument specifies the host and target architecture to use. If *architecture* isn't specified, the default build environment is used. These arguments are supported:

architecture	Compiler	Host computer architecture	Build output (target) architecture
`x86`	x86 32-bit native	x86, x64	x86
`x86_amd64` or `x86_x64`	x64 on x86 cross	x86, x64	x64
`x86_arm`	ARM on x86 cross	x86, x64	ARM
`x86_arm64`	ARM64 on x86 cross	x86, x64	ARM64
`amd64` or `x64`	x64 64-bit native	x64	x64
`amd64_x86` or `x64_x86`	x86 on x64 cross	x64	x86
`amd64_arm` or `x64_arm`	ARM on x64 cross	x64	ARM
`amd64_arm64` or `x64_arm64`	ARM64 on x64 cross	x64	ARM64

`platform_type`
This optional argument allows you to specify **`store`** or **`uwp`** as the platform type. By default, the environment is set to build desktop or console apps.

`winsdk_version`
Optionally specifies the version of the Windows SDK to use. By default, the latest installed Windows SDK is used. To specify the Windows SDK version, you can use a full Windows SDK number such as **`10.0.10240.0`**, or specify **`8.1`** to use the Windows 8.1 SDK.

`vcversion`
Optionally specifies the Visual Studio compiler toolset to use. By default, the environment is set to use the current Visual Studio compiler toolset.

Use **-vcvars_ver=14.2x.yyyyy** to specify a specific version of the Visual Studio 2019 compiler toolset.

Use **-vcvars_ver=14.29** to specify the latest version of the Visual Studio 2019 compiler toolset.

Use **-vcvars_ver=14.0** to specify the Visual Studio 2015 compiler toolset.

spectre_mode
Leave this parameter out to use libraries without Spectre mitigations. Use the value **spectre** to use libraries with Spectre mitigations.

To set up the build environment in an existing command prompt window

1. At the command prompt, use the CD command to change to the Visual Studio installation directory. Then, use CD again to change to the subdirectory that contains the configuration-specific command files. For Visual Studio 2019 and Visual Studio 2017, use the *VC\Auxiliary\Build* subdirectory. For Visual Studio 2015, use the *VC* subdirectory.

2. Enter the command for your preferred developer environment. For example, to build ARM code for UWP on a 64-bit platform, using the latest Windows SDK and Visual Studio compiler toolset, use this command line:

   ```
   vcvarsall.bat amd64_arm uwp
   ```

Create your own command prompt shortcut

Open the Properties dialog for a developer command prompt shortcut to see the command target used. For example, the target for the **x64 Native Tools Command Prompt for VS 2019** shortcut is something similar to:

```
%comspec% /k "C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\
VC\Auxiliary\Build\vcvars64.bat"
```

The architecture-specific batch files set the *architecture* parameter and call *vcvarsall.bat*. You can pass the same options to these batch files as you would pass to *vcvarsall.bat*, or you can just call *vcvarsall.bat* directly. To specify parameters for your own command shortcut, add them to the end of the command in double-quotes. For example, here's a shortcut to build ARM code for UWP on a 64-bit platform, using the latest Windows SDK. To use an earlier compiler toolset, specify the version number. Use something like this command target in your shortcut:

```
%comspec% /k "C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\
VC\Auxiliary\Build\vcvarsall.bat" amd64_arm uwp -vcvars_ver=14.29
```

Adjust the path to reflect your Visual Studio installation directory. The vcvarsall.bat file has additional information about specific version numbers.

Command-line tools

To build a C/C++ project at a command prompt, Visual Studio provides these command-line tools:

`CL`
Use the compiler (cl.exe) to compile and link source code files into apps, libraries, and DLLs.

`Link`
Use the linker (link.exe) to link compiled object files and libraries into apps and DLLs.

When you build on the command line, the F1 command isn't available for instant help. Instead, you can use a search engine to get information about warnings, errors, and messages. You can also download and use the offline help files. To use the search in Microsoft Learn, enter your query in the search box at the top of any article.

Command-line project management tools

By default, the Visual Studio IDE uses native project build systems based on MSBuild. You can invoke MSBuild directly to build projects without using the IDE. You can also use the `devenv` command to use Visual Studio to build projects and solutions. Visual Studio also supports build systems based on CMake or NMake.

`MSBuild`
Use MSBuild (msbuild.exe) and a project file (.vcxproj) to configure a build and invoke the toolset without loading the Visual Studio IDE. It's equivalent to running the **Build** project or **Build Solution** command in the Visual Studio IDE. MSBuild has advantages over the IDE when you build at the command line. You don't have to install the full IDE on all your build servers and build pipelines. You avoid the extra overhead of the IDE. MSBuild runs in containerized build environments, and supports a binary logger.

`DEVENV`
Use DEVENV (devenv.exe) combined with a command-line switch such as **`/Build`** or **`/Clean`** to execute certain build commands without displaying the Visual Studio IDE.

`CMake`
CMake (cmake.exe) is a cross-platform, open-source tool for defining build processes that run on multiple platforms. CMake can configure and control native build tools for its supported platforms, such as MSBuild and Make. For more information about CMake, see the CMake documentation.

NMAKE
Use NMAKE (nmake.exe) to build C++ projects by using a traditional makefile.

Note

Starting in Visual Studio 2019 version 16.5, MSBuild and DEVENV don't use the command-line environment to control the toolset and libraries used.

In this section

These articles show how to build apps on the command line, and describe how to customize the command-line build environment. Some show how to use 64-bit toolsets, and target x86, x64, ARM, and ARM64 platforms. They also describe use of the command-line build tools MSBuild and NMAKE.

Walkthrough: Compiling a native C++ program on the command line
Gives an example that shows how to create and compile a C++ program on the command line.

Walkthrough: Compile a C program on the command line
Describes how to compile a program written in the C programming language.

Walkthrough: Compiling a C++/CLI program on the command line
Describes how to create and compile a C++/CLI program that uses the .NET Framework.

Walkthrough: Compiling a C++/CX program on the command line
Describes how to create and compile a C++/CX program that uses the Windows Runtime.

NMAKE reference
Provides links to articles that describe the Microsoft Program Maintenance Utility (NMAKE.EXE).

MSBuild on the command line - C++
Provides links to articles that discuss how to use msbuild.exe from the command line.

Related sections

/MD, /MT, /LD (Use run-time library)
Describes how to use these compiler options to use a Debug or Release run-time library.

C/C++ compiler options
Provides links to articles that discuss the C and C++ compiler options and CL.exe.

MSVC linker options
Provides links to articles that discuss the linker options and LINK.exe.

Additional MSVC build tools
Provides links to the C/C++ build tools that are included in Visual Studio.

See also

Azure Virtual Desktop Readiness Resources | Microsoft Partner

Opportunity and Use Cases

Azure Well-Architected Azure Virtual Desktop Workload Assessment

AVD Stories

Training Resources

Azure Virtual Desktop Academy

AVD Academy Resources

Azure Virtual Desktop Landing Zone Accelerator (LZA)

Roadmap and Best Practices

AVD Community Blogs

AVD/Citrix/VMware/Azure Stack HCI Bill of Materials

AVD Level 300 Customer/Partner Deck

ur overall results

EXCELLENT

You are all set! Your results look strong and meet the necessary criteria for success.

CRITICAL 0-1 Critical: 0 to 1

MODERATE 1-2 Moderate: 1 to 2

EXCELLENT 2-3 Excellent: 2 to 3

Your result: 3/3 3 out of 3

Categories that influenced your results

Azure Virtual Desktop Readiness Resources | Microsoft Partner

EXCELLENT

You can find out how to improve on individual categories by reviewing the recommendations below in the report.

Azure Virtual Desktop | Microsoft Partner - Mar 4, 2025 - 11:55:56 AM

Your overall results	Excellent	'3/3'
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Excellent	'3/3'

Category	Link-Text	Link
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Azure Well-Architected Azure Virtual Desktop Workload Assessment	https://learn.microsoft.com/e 1e99a2021723?mode=pre-a 1e99a2021723
Azure Virtual Desktop Readiness Resources | Microsoft Partner	AVD Stories	https://azure.microsoft.com/
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Azure Virtual Desktop Academy	https://microsoft.github.io/Pa
Azure Virtual Desktop Readiness Resources | Microsoft Partner	AVD Academy Resources	https://microsoft.github.io/Pa

Azure Virtual Desktop Readiness Resources | Microsoft Partner	Azure Virtual Desktop Landing Zone Accelerator (LZA)	https://github.com/Azure/avd
Azure Virtual Desktop Readiness Resources | Microsoft Partner	AVD Community Blogs	https://techcommunity.micro
Azure Virtual Desktop Readiness Resources | Microsoft Partner	AVD/Citrix/VMware/Azure Stack HCI Bill of Materials	https://onedrive.live.com/?au
Azure Virtual Desktop Readiness Resources | Microsoft Partner	AVD Level 300 Customer/Partner Deck	https://1drv.ms/p/s!AsyILfzS8

Category	Question	Answers
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Opportunity and Use Cases	Azure Well-Architected Azure
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Opportunity and Use Cases	AVD Stories
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Training Resources	Azure Virtual Desktop Acade
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Training Resources	AVD Academy Resources
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Training Resources	Azure Virtual Desktop Landir
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Roadmap and Best Practices	AVD Community Blogs
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Roadmap and Best Practices	AVD/Citrix/VMware/Azure Sta
Azure Virtual Desktop Readiness Resources | Microsoft Partner	Roadmap and Best Practices	AVD Level 300 Customer/Par

hown below are the assessment's questions and how they were answered.

☐ Show all original responses available for each question.

Retail Readiness Resources

Messaging and Use Cases

Microsoft Cloud for Retail Partner Messaging Framework

Microsoft Cloud for Retail through-partner AI Narrative

Microsoft Retail Industry Priority Scenarios

Microsoft Cloud for Retail Reference architectures

Training Resources

Get started with Microsoft Cloud for Retail Self-Paced training

Get started with Store Operations Assist in Microsoft Cloud for Retail Self-Paced Training

Discover Microsoft AI for leaders in retail Self-Paced Training

Get started with Smart Store Analytics in Microsoft Cloud for Retail Self-Paced Training

Integrate Store Operations Assist with Microsoft Teams Self-Paced Training

---

Marketplace Offer Development Resources

Marketplace Training and Support Resources

Sell Through the Commercial Marketplace

Microsoft Commercial Marketplace Publisher FAQ

Mastering the Marketplace: Office Hours

Retail Cosell Acceleration Resources

Go-To-Market Assets & Recommended Seller Training

Retail Partner Industry Sales Kits

Microsoft Cloud for Retail Partner Assets

Recommended Sellers Training: Get started with Microsoft Cloud for Retail

---

Your overall results

EXCELLENT

You are all set! Your results look strong and meet the necessary criteria for success.

CRITICAL 0-1 Critical: 0 to 1

MODERATE 1-2 Moderate: 1 to 2

EXCELLENT 2-4 Excellent: 2 to 4

Your result: 4/4 4 out of 4

Categories that influenced your results

Retail Readiness Resources

EXCELLENT

Marketplace Offer Development Resources

EXCELLENT

Retail Cosell Acceleration Resources

EXCELLENT

You can find out how to improve on individual categories by reviewing the recommendations below in the report.

Microsoft Cloud for Retail Adoption Guide | Microsoft Partners - Mar 4, 2025 - 11:59:59 AM

Your overall results	Excellent
Retail Readiness Resources	Excellent

Marketplace Offer Development Resources	Excellent
Retail Cosell Acceleration Resources	Excellent

https://learn.microsoft.com

Category	Link-Text
Retail Readiness Resources	Microsoft Cloud for Retail Partner Messaging Framework
Retail Readiness Resources	Microsoft Cloud for Retail through-partner AI Narrative
Retail Readiness Resources	Microsoft Retail Industry Priority Scenarios
Retail Readiness Resources	Overview of Microsoft Cloud for Retail reference architectures
Retail Readiness Resources	Get started with Microsoft Cloud for Retail Self-Paced training
Retail Readiness Resources	Get started with Store Operations Assist in Microsoft Cloud for Self-Paced Training
Retail Readiness Resources	Discover Microsoft AI for leaders in retail Self-Paced Training
Retail Readiness Resources	Get started with Smart Store Analytics in Microsoft Cloud for R Self-Paced Training
Retail Readiness Resources	Integrate Store Operations Assist with Microsoft Teams Self-Pa Training
Marketplace Offer Development Resources	Sell Through the Commercial Marketplace
Marketplace Offer Development Resources	Microsoft Commercial Marketplace Publisher FAQ
Marketplace Offer Development Resources	Mastering the Marketplace: Office Hours
Retail Cosell Acceleration Resources	Retail Partner Industry Sales Kits

Retail Cosell Acceleration Resources	Microsoft Cloud for Retail Partner Assets
Retail Cosell Acceleration Resources	Recommended Sellers Training: Get started with Microsoft Clou Retail

---

Shown below are the assessment's questions and how they were answered.

☐ Show all original responses available for each question.

Sustainability Readiness Resources

Opportunity and Use Cases

This is AI ... for Sustainability

Microsoft Cloud for Sustainability: ESG and the Future of Business

---

Training Resources

Microsoft Cloud for Sustainability Training Collection

Pre-sales licenses for Sustainability Manager sandbox Environments for Microsoft Partners

Fabric Technical Resources

Microsoft Cloud for Sustainability Technical Summit

Roadmap and Best Practices

Well-Architected for Microsoft Cloud for Sustainability

What's new in Microsoft Cloud for Sustainability overview

Marketplace Offer Development Resources

Marketplace Training and Support Resources

Sell Through the Commercial Marketplace

Microsoft Commercial Marketplace Publisher FAQ

Mastering the Marketplace: Office Hours

Your overall results

EXCELLENT

You are all set! Your results look strong and meet the necessary criteria for success.

CRITICAL 0-1 Critical: 0 to 1

MODERATE 1-2 Moderate: 1 to 2

EXCELLENT 2-3 Excellent: 2 to 3

Your result: 3/3 3 out of 3

Categories that influenced your results

Sustainability Readiness Resources

EXCELLENT

Marketplace Offer Development Resources

EXCELLENT

Sustainability Cosell Acceleration Resources

EXCELLENT

You can find out how to improve on individual categories by reviewing the recommendations below in the report.

Microsoft Cloud for Sustainability Adoption Guide | Microsoft

Partners - Mar 4, 2025 - 12:03:23 PM

Your overall results	Excellent	'3/3'
Sustainability Readiness Resources	Excellent	'3/4'
Marketplace Offer Development Resources	Excellent	'3/3'
Sustainability Cosell Acceleration Resources	Excellent	'2/2'

Category	Link-Text	Link	Priority	ReportingCategory	ReportingSubcategory	Weight	Content	CompleteY/N	Note

Sustainability Readiness Resources	This is AI ... for Sustainability	https://info.microsoft.com/ww-landing-this-is-AI-for-sustainability.html?ocid=cmmop4zodtb	High	0	N
Sustainability Readiness Resources	Microsoft Cloud for Sustainability: ESG and the Future of Business	https://www.linkedin.com/learning/microsoft-cloud-for-sustainability-esg-and-the-future-of-business	High	0	N
Sustainability Readiness Resources	Microsoft Cloud for Sustainability Training Collection	https://aka.ms/mcfslearningpaths	High	0	N
Sustainability Readiness Resources	Pre-sales licenses for Sustainability Manager sandbox Environments for Microsoft Partners	https://experience.dynamics.com/requestlicense/	High	0	N
Sustainability Readiness Resources	Well-Architected for Microsoft Cloud for Sustainability	https://learn.microsoft.com/en-us/industry/well-architected/sustainability/	High	0	N
Sustainability Readiness Resources	Roadmap: What's new in Microsoft Cloud for Sustainability overview	https://learn.microsoft.com/en-us/industry/sustainability/whats-new	High	0	N
Sustainability Readiness Resources	Fabric Technical Resources	https://learn.microsoft.com/en-us/credentials/certifications/fabric-analytics-engineer-associate/?practice-assessment-	High	0	N

		type=certification			
Sustainability Readiness Resources	Microsoft Cloud for Sustainability Technical Summit	https://aka.ms/mcfstechsummit	High	0	N
Marketplace Offer Development Resources	Sell Through the Commercial Marketplace	https://learn.microsoft.com/en-us/training/paths/sell-through-commercial-marketplace/	High	0	N
Marketplace Offer Development Resources	Microsoft Commercial Marketplace Publisher FAQ	https://learn.microsoft.com/en-us/partner-center/marketplace/marketplace-faq-publisher-guide	High	0	N
Marketplace Offer Development Resources	Mastering the Marketplace: Office Hours	https://aka.ms/MTMofficehours	High	0	N
Sustainability Cosell Acceleration Resources	Sustainability Marketing Campaigns	https://partner.microsoft.com/en-us/asset/collection/sustainability-marketing-campaigns#/	High	0	N
Sustainability Cosell Acceleration Resources	Recommended Training for Sellers: Get started with Microsoft Cloud for Sustainability	https://learn.microsoft.com/en-us/training/paths/get-started-sustainability/	High	0	N

Category	Question	Answers	Selected Answer	Note
Sustainability Readiness Resources	Opportunity and Use Cases	This is AI … for Sustainability	This is AI … for Sustainability	
Sustainability Readiness Resources	Opportunity and Use Cases	Microsoft Cloud for Sustainability: ESG and the Future of Business	Microsoft Cloud for Sustainability: ESG and the Future of Business	
Sustainability Readiness Resources	Training Resources	Microsoft Cloud for Sustainability Training Collection	Microsoft Cloud for Sustainability Training Collection	
Sustainability Readiness Resources	Training Resources	Pre-sales licenses for Sustainability Manager sandbox Environments for Microsoft Partners	Pre-sales licenses for Sustainability Manager sandbox Environments for Microsoft Partners	
Sustainability Readiness Resources	Training Resources	Fabric Technical Resources	Fabric Technical Resources	
Sustainability Readiness Resources	Training Resources	Microsoft Cloud for Sustainability Technical Summit	Microsoft Cloud for Sustainability Technical Summit	
Sustainability Readiness	Roadmap and Best Practices	Well-Architected for Microsoft Cloud for Sustainability	Well-Architected for Microsoft Cloud for Sustainability	

Resources			
Sustainability Readiness Resources	Roadmap and Best Practices	What's new in Microsoft Cloud for Sustainability overview	What's new in Microsoft Cloud for Sustainability overview
Marketplace Offer Development Resources	Marketplace Training and Support Resources	Sell Through the Commercial Marketplace	Sell Through the Commercial Marketplace
Marketplace Offer Development Resources	Marketplace Training and Support Resources	Microsoft Commercial Marketplace Publisher FAQ	Microsoft Commercial Marketplace Publisher FAQ
Marketplace Offer Development Resources	Marketplace Training and Support Resources	Mastering the Marketplace: Office Hours	Mastering the Marketplace: Office Hours
Sustainability Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	Sustainability Marketing Campaigns	Sustainability Marketing Campaigns
Sustainability Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	Recommended Sellers Training: Get started with Microsoft Cloud for Sustainability	Recommended Sellers Training: Get started with Microsoft Cloud for Sustainability

Click on any field below to edit it

Page Title *experimental career

Page URL * https://archive.org/details/career-experimental-security-isc-microsoft-talk-brigth

Description * career experimental

Subject Tags * engineering

Creator tshingombe

Date 2025-02-03

Collection * Community texts

Test Item No

Language English

License Creative Commons Attribution-NonCommercial-ShareAlike

More Options Add additional metadata...

: (remove)

Drag and Drop More Files Here or

Name	Size	x
Subscriptions.csv	105 bytes	
Subscriptions(1).csv	105 bytes	
_Microsoft_Cloud_for_Retail_Adoption_Guide_Microsoft_Partners_Mar_4_2025_11_59_59_AM.csv	6.1 KB	
_Azure_Virtual_Desktop_Microsoft_Partner_Mar_4_2025_11_55_56_AM.csv	3.3 KB	
_Microsoft_Cloud_for_Sustainability_Adoption_Guide_Microsoft_Partners_Mar_4_2025_12_03_23_PM.csv	5.1 KB	
main.pdf	347 KB	
cpp-get-started-msvc-170.pdf	7.3 MB	
cpp-standard-library-msvc-170.pdf	107 MB	
isc tshingombe exam ims,, Access Control and Identity Management.docx	1.2 MB	

RF5_LT5_captions_NEW.vtt 15 KB

cpp-build-msvc-170.pdf 34 MB

Proposal of thesis content 2 final research.pdf 2.0 MB

workplace experimental ip request research met thesis aiu,,eaton microsofr eskom city power.pdf 22 MB

Gmail - ISC2 Security Congress 2025_ Call for Presentations Feedback tshingombe.pdf 2.6 MB

brighttalk-viewing-certificate-comment-créer-du-contenu-de-qualité-et-gagner-du-temps-avec-semrush.pdf 351 KB

experimental2 career thesis tlantic office pc lab(1).docx 2.9 MB

CISSP-ISSMP-Exam-Outline-November-2022-English.pdf 272 KB

CISSP-ISSEP-Exam-Outline-November-2020-English.pdf 416 KB

CISSP-ISSAP-Exam-Outline-October-2020-English.pdf 257 KB

brighttalk-viewing-certificate-lenovo-and-intel-are-driving-ai-innovation-at-th

Recommendations for your workload.

Actionable items to consider implementing to improve your workload across the five pillars of the Microsoft Azure Well-Architected Framework.

Your overall results

EXCELLENT

You are all set! Your results look strong and meet the necessary criteria for success.

CRITICAL 0-33 Critical: 0 to 33

MODERATE 33-67 Moderate: 33 to 67

EXCELLENT 67-100 Excellent: 67 to 100

Your result: 99/100 99 out of 100

Categories that influenced your results

Azure Databricks: Performance Efficiency

EXCELLENT

Data Explorer: Reliability

EXCELLENT

Azure Databricks: Operational Excellence

EXCELLENT

Azure Databricks: Cost Optimization

EXCELLENT

Azure Databricks: Security

EXCELLENT

Azure Databricks: Reliability

EXCELLENT

ADLS Gen2: Performance Efficiency

EXCELLENT

Analysis Services: Performance Efficiency

EXCELLENT

Analysis Services: Operational Excellence

EXCELLENT

Analysis Services: Cost Optimization

EXCELLENT

Analysis Services: Security

EXCELLENT

Analysis Services: Reliability

EXCELLENT

You can find out how to improve on individual categories by reviewing the recommendations below in the report.

-

Improve your results

Our recommendations for improving your results are organized by category below.

☑ ☐

☐ ☐

Azure Databricks: Performance Efficiency
EXCELLENT

1 recommended action

Data Explorer: Reliability
EXCELLENT

0 recommended actions

Azure Databricks: Operational Excellence
EXCELLENT

2 recommended actions

Azure Databricks: Cost Optimization
EXCELLENT

0 recommended actions

Azure Databricks: Security
EXCELLENT

0 recommended actions

Azure Databricks: Reliability
EXCELLENT

1 recommended action

ADLS Gen2: Performance Efficiency
EXCELLENT

0 recommended actions

Analysis Services: Performance Efficiency
EXCELLENT

0 recommended actions

Analysis Services: Operational Excellence
EXCELLENT

0 recommended actions

Analysis Services: Cost Optimization
EXCELLENT

0 recommended actions

Analysis Services: Security
EXCELLENT

0 recommended actions

Analysis Services: Reliability
EXCELLENT

0 recommended actions

- Previous Versions
- Blog
- Contribute
- Privacy
- Terms of Use
- Trademarks
- © Microsoft 2025

Recommendations for your workload.

Actionable items to consider implementing to improve your workload across the five pillars of the Microsoft Azure Well-Architected Framework.

Shown below are the assessment's questions and how they were answered.

☐ Show all original responses available for each question.

WAF analytics service selection

Which analytics service(s) are you using?

Azure Analysis Services
Azure Data Explorer
Azure Databricks
Azure Synapse Analytics
Azure Data Lake Storage Gen2

What pillars would you like to evaluate?

Reliability
Security
Cost Optimization
Operational Excellence
Performance Efficiency

Analysis Services: Reliability

What reliability targets and metrics have you defined for your application?

Azure Analysis Services RPO (Recovery Point Objective) and RTO (Recovery Time Objective) targets have been defined for the application and/or key scenarios.
Availability SLAs (Service Level Agreements) have been defined for the solution and considered while architecting the solution.
Composite SLA (including Azure Analysis Services) has been defined for the service and considered while architecting the solution.

None of the above.

How have you ensured that your application architecture is resilient to failures?

Backup databases from primary server can be restored on redundant servers.
An alias is used for the primary server to avoid having to change the connection strings on reporting clients.
The database can be restored using SSMS or PowerShell using backups located in a storage account configured for the server.
We use asynchronous refresh with the REST API to prevent the need for long-running HTTP connections, auto retries, and batched commits.
None of the above.

How have you ensured required capacity and services are available in targeted regions?

We use the Azure Analysis Services resource and object limits as part of the resource governance mechanisms to enforce these limits.
We verify that the Azure Analysis Services product is available in the target region in case that region is not the paired-region.
We verify that the target region meets regulatory or governance requirements for the data stored in the data lake (HIPAA, PCI, etc.)
None of the above.

How are you handling disaster recovery for this workload?

We consider the incidents we want to be protected from when choosing the right redundancy options, and build the backup and disaster recovery plan accordingly: local outage, regional disaster, capacity limitations, and so on.
We deploy models to redundant servers in other regions.
We complete periodic disaster recovery exercises to ensure the procedure works as planned.
We verify end-to-end system performance after failover occurs.
None of the above.

How do you monitor and measure workload health?

We use the available metrics, logs, and diagnostics with Azure Monitor.
We use Azure Monitor metric alerts with dynamic thresholds detection.

We use Azure Resource Health events to alert on resource health events.
We use Azure Service Health events to alert on applicable service-level events (service issue, planned maintenance, health advisories, and security advisory).
None of the above.

Analysis Services: Security

What design considerations did you make in your workload with regard to security?

We identify and classify business-critical datasets that might adversely affect operations if they're compromised or become unavailable.
We list and document security requirements for the Azure Analysis Services instances.
We limit access by following the principle of least privilege, protecting data, and monitoring activities offered by Azure Analysis Services.
We define and test operational processes for incident response.
We review data compliance requirements and choose the appropriate regulatory compliance controls for the data.
None of the above.

What considerations for compliance and governance do you need to take?

We use Azure policies to enforce security, compliance, and organizational standards.
We activate diagnostic logs and store them in a log analytics workspace.
None of the above.

How are you managing encryption for this workload?

We rotate the storage key by specifying a key expiration period.
We compress and encrypt tabular model backups.
None of the above.

How are you managing permissions for this workload?

User identities are registered within the Microsoft Entra tenant of the subscription.
We use Azure Analysis Services role-based access control (RBAC).

We use object-level security, which includes table-level security and column-level security.
We implement row-level security as part of tabular model roles.
None of the above.

How have you secured the network of your workload?

We configured a server firewall to filter the inbound traffic.
We use gateways for on-premises data sources.
We use gateways for data sources on Azure Virtual Networks.
We use DirectQuery mode, where only metadata is stored.
None of the above.

Analysis Services: Cost Optimization

What actions are you taking to optimize cloud costs?

We're getting familiar with Analysis Services pricing, in particular, with elements that drive the price calculation: QPUs, memory (GB), presence of SLA, scale-out instances, tier, uptime, region.
We use the Cost Management and Billing tools to analyse and manage costs.
Our choice of region in which to deploy the instance in is driven by the data source’s location, the users’ location, the availability of the SKU and of the query replica feature (if it's planned to be used), and the price.
We choose the right tier and plan for the instance.
We use the right upgrade plan that will allow for more query processing units and more cache capacity.
We pause the server when not in use, and resume when needed to only pay for what is used.
As an existing Power BI Premium Capacity user, we evaluate the use of Power BI Premium that embeds an Analysis Services instance instead of a standalone Azure Analysis Services server.
None of the above.

How is your organization modeling cloud costs?

We use the cost control feature to set the budget as part of Cost Management for Azure Analysis Services.
We lower costs by pausing or scaling the instance to meet performance demands.
None of the above.

How are you monitoring your costs?

We use the cost analysis feature of Azure Cost Management to monitor costs and create budgets.
None of the above.

Analysis Services: Operational Excellence

How are you designing your applications to take into account DevOps?

We use source-control integration as the first step in building a continuous integration and deployment pipeline.
We use continuous integration and deployment for Azure Analysis Services, and build continuous integration and deployment pipeline for data models.
None of the above.

How are you managing the configuration of your workload?

We automate creation of Azure Analysis Services instances with ARM templates, PowerShell scripts, or Bicep files.
We use Dynamic Management Views (DMVs) in Analysis Services to monitor server instances.
We monitor server metrics provided by Azure Monitor (memory, CPU usage, number of client connections, query resource consumption, etc.).
We enable diagnostic logging for Azure Analysis Services to monitor and send logs to Azure storage, stream them to Azure Event Hubs, or export them to Azure Monitor logs.
We add resource health alerts for metrics such as memory usage, memory limit High, and memory limit Hard.
None of the above.

What operational considerations are you making regarding the deployment of your workload?

We're becoming familiar with Azure Analysis Services resource and object limits to learn what happens when those resource limits are hit or exceeded, and describe the resource governance mechanisms used to enforce these limits.

No routine and manual operational changes are performed outside of IaC (Infrastructure as Code) to prevent configuration drift by enforcing consistency representing desired environment states.
Critical test environments have 1:1 parity with the production environment.
None of the above.

What processes and procedures have you adopted to optimize workload operability?

Specific methodologies, like DevOps, are used to structure the development and operations process.
We leverage central Azure monitoring tools like Azure Monitor.
Data analysts, data engineers, development teams, and the operations team collaborate to resolve production issues that are clearly defined and well understood.
Operational shortcomings and failures are analyzed, post-mortems are performed and used to improve and refine operational procedures.
There are tools or processes in place, such as Microsoft Entra Privileged Identity Management, to grant access to critical instances on a just-in-time basis.
Azure Resource Tags are used to enrich our AAS instance with operational metadata.
There are tools and processes, like Azure Policy, in place to govern available services, enforce mandatory operational functionality and ensure compliance.

How are you monitoring for a healthy workload?

We use Azure Monitor to perform more in-depth diagnostics, track performance, and identify trends using the platform metrics.
We use Azure Diagnostics to offload Platform logs.
We use Extended Events.
We use Dynamic Management Views (DMVs).

Analysis Services: Performance Efficiency

How are you designing your workload to scale?

We maintain a healthy workload by automating the scaling of the Azure Analysis Services instances.
We use read-only replicas scale-out to have queries return consistent data while processing data.

We separate the processing server from the query pool to ensure that client queries aren't affected by processing operations.
None of the above.

How are you handling user load?

We create a query pool with up to seven additional query replicas (eight total, including the server).
We simplify the query or its calculations if the query is too memory intensive.

How are you ensuring that you have sufficient memory?

We monitor datasets to not exceed available server resource memory.
We monitor the memory usage broken out by database.
None of the above.

How are you managing your data to handle scale?

We use partitioning to take advantage of incremental loads.
We refresh (process) in-memory models to update cached data from data sources.
We keep the data model as simple as possible by removing unneeded columns and keeping the size to the minimum, paying attention to the data types.
None of the above.

How are you monitoring to ensure the workload is scaling appropriately?

With Azure service principal support, we perform unattended refresh operations using PowerShell, TOM, TMSL, or REST to make sure our model data is always up to date.
We implement asynchronous refresh with the REST APIs to mitigate long-running operations.
None of the above.

Data Factory: Reliability

Data Factory: Security

Data Factory: Cost Optimization

Data Factory: Operational Excellence

Data Factory: Performance Efficiency

Azure Databricks: Reliability

How do you implement reliability Best Practices?

We deploy workspaces in multiple subscriptions based on service limits, including Databricks workspace limits and Azure subscription limits.
We leverage clusters pools with TTL=60 min, or interactive clusters for job-based scenarios where we expect to spin up or down quickly.
We stagger job-based clusters in the same workspace for scenarios requiring quick spin up and down of job clusters at less than 5 minutes as the recommended interval.
We use Cluster Scoped Init scripts rather than global or named scripts.
We ensure that we've configured an appropriate level of data redundancy for our use case.
None of the above.

How do you implement disaster recovery scenarios?

We disable RA-GRS stores in development subscriptions to reduce cost.
We use RA-GRS storage accounts only when required to meet disaster planning.
We enable soft delete, snapshot, and point in time recovery (PITR) for storage.
We perform daily backups of Databricks configuration.
We use the cluster log delivery feature to manage logs.

Azure Databricks: Security

How do you implement security Best Practices?

We store any production data in default Azure Databricks file system (DBFS) folders.
We deploy the Databricks workspace in our virtual network.
We review and plan to implement controls in Microsoft Blueprints for HiTrust / HIPAA and PCI/DSS.

We have a process in place to periodically regenerate our account keys.
We implement a security development lifecycle and threat model to assess risks in our application.
We enable advanced threat protection for storage.

How do you implement authentication controls?

We enable access control lists to configure permissions at the workspace, clusters, pools, jobs, and data tables.
We use credential passthrough to authenticate automatically.
We use Azure Key Vault (AKV) to store secrets, including credentials.
We enable customer manage keys (CMK) for notebooks and root Databricks File System (DBFS).
We enable Oauth authentication.

How do you implement encryption on your clusters?

We encrypt traffic between clusters and worker nodes.
We set up a minimum transport layer security (TLS) version for all storage accounts to TLS 1.2.
We limit shared access signature (third-party tools) tokens to HTTPS connections only.
None of the above.

How do you implement security at the networking level?

We enable IP access lists to restrict access to certain IP addresses.
We limit private IP addresses.
We leverage Azure Private Endpoint.
We leverage No Public IP (NPIP).
We safelist service principals' names and personal access tokens.
We enable virtual network (Vnet) injection.
None of the above.

How do you audit and monitor your Databricks platform for security?

We enable audit logging.
We ingest log data into a security information and event manager (SIEM) for security monitoring.
We review and reconcile user access.

Azure Databricks: Cost Optimization

How do you implement cost optimization Best Practices?

Users can share autoscaling clusters rather than each user having to create a separate cluster.
We leverage the right SKU for the scenario, that is, Jobs Compute for data engineering and Batch ELT workload with single Jobs Compute cluster.
We use chargeback scenarios.
We review file formats and compute and network and identify areas for cost optimization.
We regularly use the delta optimizer to merge small files into larger files.
None of the above.

How do you implement cost savings?

We pre-purchase commit units and reserve VM instances when possible.
We choose Azure regions that offer the lowest cost while meeting performance requirements.
None of the above.

How do you monitor Azure Databricks costs?

We monitor costs of clusters using the cost analysis report.
We set up budget alerts to monitor costs.
We use Databricks Overwatch.
None of the above.

Azure Databricks: Operational Excellence

How do you implement operational Best Practices?

We do regular performance, scalability, and stress testing.
We build a process to review Azure Advisor and Azure Security Center recommendations on a regular cadence.
We review and address platform changes from the release notes.
We split workspaces for Dev, QA, and Production.
We use automated clusters for production jobs instead of interactive clusters.
We run auto-optimization to improve performance for the downstream.
We optimize and curate delta tables (silver tables).

We review the continuous integration and continuous deployment (CI/CD) automation framework.
We run monthly log reviews to validate environment health.
We terminate and rebuild clusters on a frequent basis to ensure Databricks clusters are patched by Microsoft.

How do you monitor your Databricks platform for operations?

We enable logging and alerting for all components in the Databricks platform.
We use dashboards to visualize metrics.
We set up any cluster activity monitoring.
We enable storage account logging.
We put a single point of log aggregation in place.
We use Network Watcher to collect and Monitor network activity.
We ensure that all application-level monitoring is enabled.
We use a single pane of glass with telemetry using Log Analytics logs to EventHub for consumption by other systems.
We consider ingesting selected logs from Azure storage accounts to Azure Monitor.
We monitor for 500 errors by Databricks, Blob storage, or other HTTP endpoints.
We implement cluster secure management.

What components of your Azure environment do you monitor as part of your operations practice?

We use approved time synchronization sources.
We configure central security log management.
We enable audit logging for Azure resources.
We collect security logs from operating systems.
We configure security log storage retention.
We monitor and review logs.
We enable alerts for anomalous activities.
We centralize anti-malware logging.
We enable DNS query logging.
We enable command-line audit logging.

Azure Databricks: Performance Efficiency

How do you implement performance Best Practices?

We choose the correct cluster size by doing iterative performance testing.
We regularly conduct rigorous quality and unit testing to validate performance that meets requirements.
We leverage the auto-scaling feature with auto-terminate.
We turn shuffle off for optimal performance.
We check for data skew.
We ensure that the file size and format are homogenous.
We consistently use DataFrame API and SparkSQL.
We avoid user-defined functions (UDFs) , especially in Python or R.
We consider and test repartitioning if we need to join large tables.
We ensure Azure limits are increased, for example, Public IP limits and so on.

How do you optimize performance efficiency?

We reorder skew joins.
We optimize for performance with Delta Lake format to get the best price to performance ratio.
We partition our data.
We check for large shuffle joins and try replacing them with broadcasts.
We use Delta Lake with Z-order and optimize the latest Databricks Runtime (DBR) to get the best performance.
We use Parquet file format.
We use Delta-Cache.

How do you test performance efficiency on the Azure Databricks clusters?

We run a proof of concept to determine how often to execute based on data ingestion and query patterns.
We engage with Azure Engineers to ensure that capacity can be handled in the backend and limits get increased.
We engage with the networking team during testing.
We ensure throttling is not hit by setting up Azure Data Lake Storage Gen 2 limits.
We review all Azure and Databricks limits.
We develop a medium-sized cluster of 2-8 nodes, with VMs matched to the workload class, as explained earlier.
We run end to end tests on larger representative data while measuring CPU, memory, and I/O used by the cluster at an aggregate level.
We optimize the cluster to remove bottlenecks.

How do you monitor your Databricks platform for performance efficiency?

We troubleshoot performance bottlenecks by using dashboards to identify job and stage latency and streaming throughput.
We validate whether upstream components can sustain the load required to pass through them.
We run scheduled optimization on delta tables.
We tune shuffle for optimal performance.
We use autoscaling methodologies whenever possible.
We partition our data following Best Practices.
None of the above.

How do you support interactive analytics using shared high-concurrency clusters?

We deploy a shared cluster instead of letting each user create their cluster.
We create the shared cluster in High Concurrency mode instead of Standard mode.
We configure security on the shared high concurrency cluster.
None of the above.

Data Explorer: Reliability

What reliability targets and metrics have you defined for your application?

Ensure that the average CPU is running at 80% capacity or less and cache utilization is 100%.
Use Resource Health to monitor the status of Azure Data Explorer.

How have you ensured that your application architecture is resilient to failures?

This question was left unanswered

How are you handling disaster recovery for this workload?

This question was left unanswered

How do you monitor and measure application health?

This question was left unanswered

Data Explorer: Security

What action are you taking to meet your compliance and governance requirements?

This question was left unanswered

How are you protecting data for this workload?

This question was left unanswered

How are you managing identity for this workload?

This question was left unanswered

How do you keep your Azure Data Explorer cluster from being exposed to the internet?

This question was left unanswered

Data Explorer: Cost optimization

What actions are you taking to optimize cloud costs?

This question was left unanswered

How do you ensure that cloud resources are appropriately provisioned?

This question was left unanswered

How is your organization modeling cloud costs?

This question was left unanswered

How do you manage the storage footprint of your digital assets?

This question was left unanswered

How are you monitoring your costs?

This question was left unanswered

What tradeoffs have you made to optimize for cost?

This question was left unanswered

Data Explorer: Operational excellence

How are you designing your applications to take DevOps into account?

This question was left unanswered

How are you managing the configuration of your workload?

This question was left unanswered

What considerations are you making around the deployment of your infrastructure?

This question was left unanswered

Are you using best practices for Kusto queries?

This question was left unanswered

How are you monitoring your deployments and workload?

This question was left unanswered

Data Explorer: Performance efficiency

How are you designing your workload to scale?

This question was left unanswered

How do you optimize Azure Data Explorer workloads for performance?

This question was left unanswered

How are you ensuring that you have sufficient capacity?

This question was left unanswered

How are you monitoring to ensure the workload is scaling appropriately?

This question was left unanswered

Synapse: Reliability

What reliability targets and metrics have you defined for your application?

This question was left unanswered

How have you ensured that your application architecture is resilient to failures?

This question was left unanswered

How have you ensured required capacity and services are available in targeted regions?

This question was left unanswered

How are you handling disaster recovery for this workload?

This question was left unanswered

What decisions have been taken to ensure the application platform meets your reliability requirements?

This question was left unanswered

How does your application logic handle exceptions and errors?

This question was left unanswered

What decisions have been taken to ensure networking and connectivity meets your reliability requirements?

This question was left unanswered

What reliability allowances for scalability and performance have you made?

This question was left unanswered

What reliability allowances for security have you made?

This question was left unanswered

What reliability allowances for operations have you made?

This question was left unanswered

How do you test the application to ensure it is fault tolerant?

This question was left unanswered

How do you monitor and measure application health?

This question was left unanswered

Synapse: Security

What design considerations did you make in your workload in regards to security?

This question was left unanswered

What considerations for compliance and governance do you need to take?

This question was left unanswered

How are you managing encryption for this workload?

This question was left unanswered

How are you managing identity for this workload?

This question was left unanswered

How have you secured the network of your workload?

This question was left unanswered

What tradeoffs do you need to make to meet your security goals?

This question was left unanswered

How are you ensuring your critical accounts are protected?

This question was left unanswered

Synapse: Cost Optimization

What actions are you taking to optimize cloud costs?

This question was left unanswered

How do you ensure that cloud resources are appropriately provisioned?

This question was left unanswered

How is your organization modeling cloud costs?

This question was left unanswered

How do you manage the storage footprint of your digital assets?

This question was left unanswered

How are you monitoring your costs?

This question was left unanswered

What trade-offs have you made to optimize for cost?

This question was left unanswered

Synapse: Operational Excellence

How are you designing your applications to take into account DevOps?

This question was left unanswered

How are you managing the configuration of your workload?

This question was left unanswered

What considerations are you making around the deployment of your infrastructure?

This question was left unanswered

How is development done on this workload?

This question was left unanswered

How are you monitoring your deployments and workload?

This question was left unanswered

How are you integrating your workloads?

This question was left unanswered

Synapse: Performance Efficiency

How are you designing your workload to scale?

This question was left unanswered

How do you optimize Synapse workloads for performance?

This question was left unanswered

How are you ensuring you have sufficient capacity?

This question was left unanswered

How are you managing your data to handle scale?

This question was left unanswered

How are you monitoring to ensure the workload is scaling appropriately?

This question was left unanswered

ADLS Gen2: Reliability

How have you ensured that your application architecture is resilient to failures?

This question was left unanswered

How have you ensured required capacity and services are available in targeted regions?

This question was left unanswered

How are you handling disaster recovery for this workload?

This question was left unanswered

How does your application logic handle exceptions and errors?

This question was left unanswered

What decisions have been taken to ensure networking and connectivity meets your reliability requirements?

This question was left unanswered

How do you monitor and measure application health?

This question was left unanswered

How do you mitigate accidental deletion of your data?

This question was left unanswered

How do you ensure the availability of your most critical datasets?

This question was left unanswered

How do you ensure that the data is reliable?

This question was left unanswered

ADLS Gen2: Security

What design considerations did you make in your workload with regard to security?

This question was left unanswered

What considerations for compliance and governance do you need to take?

This question was left unanswered

How are you managing encryption for this workload?

This question was left unanswered

How are you managing identity and authorization for this workload?

This question was left unanswered

How have you secured the network of your workload?

This question was left unanswered

What tradeoffs do you need to make to meet your security goals?

This question was left unanswered

ADLS Gen2: Cost Optimization

What actions are you taking to optimize cloud costs?

This question was left unanswered

How are you monitoring your costs?

This question was left unanswered

ADLS Gen2: Operational Excellence

What tooling do you leverage to monitor your ADLS accounts?

This question was left unanswered

What considerations are you making around the deployment of your workload?

This question was left unanswered

How are you integrating your workloads?

This question was left unanswered

What processes and procedures have you adopted to optimize workload operability?

This question was left unanswered

Has the data been organized in the data lake to optimize for access, performance, and usability?

This question was left unanswered

How do you make your data discoverable for users?

This question was left unanswered

ADLS Gen2: Performance Efficiency

How are you designing your workload to scale?

This question was left unanswered

How do you optimize ADLS workloads for performance?

This question was left unanswered

How are you managing your data to handle scale?

This question was left unanswered

How are you ensuring you have sufficient capacity?

Review ADLS Gen2 product limits.
Monitor ADLS Gen2 resource utilization, query activity, and other metrics that have limitations.
None of the above

- Previous Versions
- Blog
- Contribute
- Privacy
- Terms of Use
- Trademarks
- © Microsoft 2025

our overall results

LOW

Room to improve. It looks like there are key items needing attention. Review the recommendations to see what actions you can take to improve your results.

LOW 0-12 Low: 0 to 12

MODERATE 12-23 Moderate: 12 to 23

EXCELLENT 23-35 Excellent: 23 to 35

Your result: 10/35 10 out of 35

Categories that influenced your results

Azure AI Fundamentals

LOW

Designing and Implementing a Microsoft Azure AI Solution

MODERATE

You can find out how to improve on individual categories by reviewing the recommendations below in the report.

Azure AI Fundamentals

Fundamental AI Concepts

Azure AI Bot Service

Fundamentals of question answering with the Language Service

"Create an empty knowledge base, and then manually copy and paste the FAQ entries into it.

Fundamentals of Azure AI Document Intelligence

Azure AI Vision resource

Fundamentals of Azure OpenAI Service

Azure OpenAI is Microsoft's version of ChatGPT, a chatbot that uses generative AI models.

Designing and Implementing a Microsoft Azure AI Solution

Prepare to develop AI solutions on Azure

Absolutely correct values based on conditional logic.

Secure Azure AI services

Switch the app to use the secondary key

Deploy Azure AI services in containers

Client applications must pass a subscription key to the Azure resource endpoint before using the container.

Make recommendations with Azure AI Personalizer

In the Azure portal, go to the Monitor page for your AI Personalizer resource, and view the Personalizer average reward.

Analyze images

Tags

Classify images

"Image classification (multiclass)

Detect, analyze, and recognize faces

Location

Analyze video

Use the Azure AI Vision service to extract key frames from the video.

Build a question answering solution

Create an empty knowledge base and manually enter the FAQ questions and answers.

Build a conversational language understanding model

Intents

Develop an app with Azure AI Language

Sentiment analysis

Create a custom text classification solution

A multiple label classification project

Create a custom named entity extraction solution

Recall

Translate text with the Azure AI Translator service

Detect

Create speech-enabled apps with Azure AI services

The location and one of the keys

Translate speech with the Azure AI Speech service

SpeechConfig

Create an Azure Cognitive Search solution

Add a JSON file that defines an Azure AI Search index to the blob container.

Create a custom skill for Azure Cognitive Search

Create a custom skill that uses an Azure Machine Learning model to predict the sentiment for a document.

Create a knowledge store with Azure Cognitive Search

Merge

Enrich a search index using Language Studio

Conversational language understanding.

Implement advanced search features in Azure Cognitive Search

^

Build an Azure Machine Learning custom skill for Azure Cognitive Search

Real-time endpoint

Maintain an Azure Cognitive Search Solution

Create an Azure Cognitive Search service with a Storage Optimized service tier and at least two replicas.

Use semantic search to get better search results in Azure Cognitive Search

As many results as the BM25 ranking function returns.

Improve search results using vector search in Azure Cognitive Search

To create a search to match text input.

Plan an Azure AI Document Intelligence solution

A Composed model.

Use prebuilt Azure AI Document Intelligence models

Read model.

Create a composed Form Recognizer model

modelId

Generate code with Azure OpenAI Service

Increase in efficiency and productivity

Generate images with Azure OpenAI Service

GPT-35-Turbo

Fundamentals of Responsible Generative AI

To make a legal case that indemnifies you from responsibility for

AI Engineer Skill Assessment - Mar 4, 2025 - 1:05:49 PM		
Your overall results	Low	'10/35'
Azure AI Fundamentals	Low	'0/4'
Designing and Implementing a Microsoft Azure AI Solution	Moderate	'10/31'

Category	Link-Text	Link
Azure AI Fundamentals		https://learn.microsoft.com
Azure AI Fundamentals	Microsoft Azure AI Fundamentals: Computer Vision	https://learn.microsoft.com/e vision-microsoft-azure/
Azure AI Fundamentals	Microsoft Azure AI Fundamentals: Natural Language Processing	https://learn.microsoft.com/e language-processing/
Azure AI Fundamentals	Microsoft Azure AI Fundamentals: Document Intelligence and Knowledge Mining	https://learn.microsoft.com/e intelligence-knowledge-mini
Azure AI Fundamentals	Microsoft Azure AI Fundamentals: Generative AI	https://learn.microsoft.com/e generative-ai/
Designing and Implementing a Microsoft Azure AI Solution		https://learn.microsoft.com
Designing and Implementing a Microsoft Azure AI Solution	Get started with Azure AI Services	https://learn.microsoft.com/e
Designing and Implementing a Microsoft Azure AI Solution	Develop decision support solutions with Azure AI Services	https://learn.microsoft.com/e support/
Designing and Implementing a Microsoft Azure AI	Create computer vision solutions with Azure AI Vision	https://learn.microsoft.com/e vision-solutions-azure-ai/

Solution		
Designing and Implementing a Microsoft Azure AI Solution	Develop natural language processing solutions with Azure AI Services	https://learn.microsoft.com/e solutions-azure-ai/
Designing and Implementing a Microsoft Azure AI Solution	Implement knowledge mining with Azure Cognitive Search	https://learn.microsoft.com/e knowledge-mining-azure-cog
Designing and Implementing a Microsoft Azure AI Solution	Develop solutions with Azure AI Document Intelligence	https://learn.microsoft.com/e forms-document-intelligence
Designing and Implementing a Microsoft Azure AI Solution	Develop Generative AI solutions with Azure OpenAI Service	https://learn.microsoft.com/e azure-openai/

Category	Question	Answers
Azure AI Fundamentals	Fundamental AI Concepts	Azure Machine Learning
Azure AI Fundamentals	Fundamental AI Concepts	Azure AI Bot Service
Azure AI Fundamentals	Fundamental AI Concepts	Azure AI Language
Azure AI Fundamentals	Fundamentals of machine learning	Regression
Azure AI Fundamentals	Fundamentals of machine learning	Classification
Azure AI Fundamentals	Fundamentals of machine learning	Clustering
Azure AI Fundamentals	Fundamentals of Azure AI services	A multi-service resource tha
Azure AI Fundamentals	Fundamentals of Azure AI services	A single-service resource for
Azure AI Fundamentals	Fundamentals of Azure AI services	It's not possible to see costs
Azure AI Fundamentals	Fundamentals of Computer Vision	Timestamps in photograph r
Azure AI Fundamentals	Fundamentals of Computer Vision	Pixels
Azure AI Fundamentals	Fundamentals of Computer Vision	Image file names
Azure AI Fundamentals	Fundamentals of Facial Recognition	A pair of coordinates for eac
Azure AI Fundamentals	Fundamentals of Facial Recognition	Two pairs of coordinates for eyes
Azure AI Fundamentals	Fundamentals of Facial Recognition	A set of coordinates for each around the face
Azure AI Fundamentals	Fundamentals of optical character recognition	Azure AI Vision
Azure AI Fundamentals	Fundamentals of optical character recognition	Azure AI services
Azure AI Fundamentals	Fundamentals of optical character recognition	Azure AI Language
Azure AI Fundamentals	Fundamentals of Text Analysis with the Language Service	Sentiment analysis
Azure AI Fundamentals	Fundamentals of Text Analysis with the Language Service	Key phrase extraction
Azure AI Fundamentals	Fundamentals of Text Analysis with the	Entity detection

	Language Service	
Azure AI Fundamentals	Fundamentals of question answering with the Language Service	"Create an empty knowledge the FAQ entries into it.
Azure AI Fundamentals	Fundamentals of question answering with the Language Service	Import the existing FAQ docu
Azure AI Fundamentals	Fundamentals of question answering with the Language Service	Import a pre-defined chit-cha
Azure AI Fundamentals	Fundamentals of conversational language understanding	Azure AI Speech
Azure AI Fundamentals	Fundamentals of conversational language understanding	Azure AI Language
Azure AI Fundamentals	Fundamentals of conversational language understanding	Azure AI services
Azure AI Fundamentals	Fundamentals of Azure AI Speech	Speech
Azure AI Fundamentals	Fundamentals of Azure AI Speech	Language
Azure AI Fundamentals	Fundamentals of Azure AI Speech	Azure AI services
Azure AI Fundamentals	Fundamentals of Azure AI Document Intelligence	Azure AI Vision resource
Azure AI Fundamentals	Fundamentals of Azure AI Document Intelligence	Azure AI Document Intelligen
Azure AI Fundamentals	Fundamentals of Azure AI Document Intelligence	Azure AI Language resource
Azure AI Fundamentals	Fundamentals of Knowledge Mining with Azure Cognitive Search	CSV
Azure AI Fundamentals	Fundamentals of Knowledge Mining with Azure Cognitive Search	SQL
Azure AI Fundamentals	Fundamentals of Knowledge Mining with Azure Cognitive Search	JSON
Azure AI Fundamentals	Fundamentals of Generative AI	Models that only work with c
Azure AI Fundamentals	Fundamentals of Generative AI	Models that only work with s
Azure AI Fundamentals	Fundamentals of Generative AI	Models that use deep learnin language on a massive scale
Azure AI Fundamentals	Fundamentals of Azure OpenAI Service	Azure OpenAI is Microsoft's v generative AI models.
Azure AI Fundamentals	Fundamentals of Azure OpenAI Service	ChatGPT and OpenAI are cha code, and images. Azure Op chatbots.
Azure AI Fundamentals	Fundamentals of Azure OpenAI Service	OpenAI is a research compan uses generative AI models. A OpenAI's AI models.
Azure AI Fundamentals	Fundamentals of Responsible Generative AI	To make a legal case that in harms caused by the solutio
Azure AI Fundamentals	Fundamentals of Responsible Generative AI	To document the purpose, e solution
Azure AI Fundamentals	Fundamentals of Responsible Generative AI	To evaluate the cost of cloud solution
Designing and Implementing a Microsoft Azure AI Solution	Prepare to develop AI solutions on Azure	Absolutely correct values ba

Designing and Implementing a Microsoft Azure AI Solution	Prepare to develop AI solutions on Azure	Randomly selected values w
Designing and Implementing a Microsoft Azure AI Solution	Prepare to develop AI solutions on Azure	Probabilistic values based or
Designing and Implementing a Microsoft Azure AI Solution	Create and consume Azure AI services	The application must specify resource.
Designing and Implementing a Microsoft Azure AI Solution	Create and consume Azure AI services	The user of the application n associated with the Azure su
Designing and Implementing a Microsoft Azure AI Solution	Create and consume Azure AI services	Access to Azure AI Services
Designing and Implementing a Microsoft Azure AI Solution	Secure Azure AI services	Switch the app to use the se
Designing and Implementing a Microsoft Azure AI Solution	Secure Azure AI services	Change the resource endpoi
Designing and Implementing a Microsoft Azure AI Solution	Secure Azure AI services	Enable a firewall
Designing and Implementing a Microsoft Azure AI Solution	Monitor Azure AI services	Create an alert.
Designing and Implementing a Microsoft Azure AI Solution	Monitor Azure AI services	Configure diagnostic setting
Designing and Implementing a Microsoft Azure AI Solution	Monitor Azure AI services	Create a dashboard.
Designing and Implementing a Microsoft Azure AI Solution	Deploy Azure AI services in containers	Client applications must pas endpoint before using the co
Designing and Implementing a	Deploy Azure AI services in containers	The container must be able t to send usage data for billing

Microsoft Azure AI Solution		
Designing and Implementing a Microsoft Azure AI Solution	Deploy Azure AI services in containers	All data passed from the clie forwarded to the Azure reso
Designing and Implementing a Microsoft Azure AI Solution	Make recommendations with Azure AI Personalizer	In the Azure portal, go to the resource, and view the Perso
Designing and Implementing a Microsoft Azure AI Solution	Make recommendations with Azure AI Personalizer	In the Azure portal, go to the resource, and view the Base
Designing and Implementing a Microsoft Azure AI Solution	Make recommendations with Azure AI Personalizer	In the Azure portal, go to the resource, and view the Rewa
Designing and Implementing a Microsoft Azure AI Solution	Analyze images	Tags
Designing and Implementing a Microsoft Azure AI Solution	Analyze images	Description
Designing and Implementing a Microsoft Azure AI Solution	Analyze images	Categories
Designing and Implementing a Microsoft Azure AI Solution	Classify images	"Image classification (multic
Designing and Implementing a Microsoft Azure AI Solution	Classify images	Image classification (multila
Designing and Implementing a Microsoft Azure AI Solution	Classify images	Object detection
Designing and Implementing a Microsoft Azure AI Solution	Detect objects in images	The location and class of spe
Designing and Implementing a Microsoft Azure AI Solution	Detect objects in images	The class of the main subjec

Designing and Implementing a Microsoft Azure AI Solution	Detect objects in images	The file type of an image.
Designing and Implementing a Microsoft Azure AI Solution	Detect, analyze, and recognize faces	Location
Designing and Implementing a Microsoft Azure AI Solution	Detect, analyze, and recognize faces	Type of eye-glasses
Designing and Implementing a Microsoft Azure AI Solution	Detect, analyze, and recognize faces	Occlusion
Designing and Implementing a Microsoft Azure AI Solution	Read Text in Images and Documents with the Azure AI Vision Service	Only total content and page
Designing and Implementing a Microsoft Azure AI Solution	Read Text in Images and Documents with the Azure AI Vision Service	Pages, words and lines of te
Designing and Implementing a Microsoft Azure AI Solution	Read Text in Images and Documents with the Azure AI Vision Service	Total content, pages, words
Designing and Implementing a Microsoft Azure AI Solution	Analyze video	Use the Azure AI Vision serv
Designing and Implementing a Microsoft Azure AI Solution	Analyze video	Upload the video to Azure V
Designing and Implementing a Microsoft Azure AI Solution	Analyze video	Store the video file in an Azu
Designing and Implementing a Microsoft Azure AI Solution	Analyze text with Azure AI Language	Use the Azure AI Language s
Designing and Implementing a Microsoft Azure AI Solution	Analyze text with Azure AI Language	Use the Azure AI Language s the comments.
Designing and Implementing a	Analyze text with Azure AI Language	Use the Azure AI Language s comments.

Microsoft Azure AI Solution		
Designing and Implementing a Microsoft Azure AI Solution	Build a question answering solution	Create an empty knowledge questions and answers.
Designing and Implementing a Microsoft Azure AI Solution	Build a question answering solution	Create a new knowledge bas
Designing and Implementing a Microsoft Azure AI Solution	Build a question answering solution	"Create a new knowledge ba chat source.
Designing and Implementing a Microsoft Azure AI Solution	Build a conversational language understanding model	Intents
Designing and Implementing a Microsoft Azure AI Solution	Build a conversational language understanding model	Utterances
Designing and Implementing a Microsoft Azure AI Solution	Build a conversational language understanding model	Entities
Designing and Implementing a Microsoft Azure AI Solution	Develop an app with Azure AI Language	Sentiment analysis
Designing and Implementing a Microsoft Azure AI Solution	Develop an app with Azure AI Language	Key phrase extraction
Designing and Implementing a Microsoft Azure AI Solution	Develop an app with Azure AI Language	Entity recognition
Designing and Implementing a Microsoft Azure AI Solution	Create a custom text classification solution	A single label classification p
Designing and Implementing a Microsoft Azure AI Solution	Create a custom text classification solution	A multiple label classificatio
Designing and Implementing a Microsoft Azure AI Solution	Create a custom text classification solution	A varied label classification p

Designing and Implementing a Microsoft Azure AI Solution	Create a custom named entity extraction solution	Recall
Designing and Implementing a Microsoft Azure AI Solution	Create a custom named entity extraction solution	Precision
Designing and Implementing a Microsoft Azure AI Solution	Create a custom named entity extraction solution	F1 score
Designing and Implementing a Microsoft Azure AI Solution	Translate text with the Azure AI Translator service	Detect
Designing and Implementing a Microsoft Azure AI Solution	Translate text with the Azure AI Translator service	Translate
Designing and Implementing a Microsoft Azure AI Solution	Translate text with the Azure AI Translator service	Transliterate
Designing and Implementing a Microsoft Azure AI Solution	Create speech-enabled apps with Azure AI services	The location and one of the
Designing and Implementing a Microsoft Azure AI Solution	Create speech-enabled apps with Azure AI services	The primary and secondary
Designing and Implementing a Microsoft Azure AI Solution	Create speech-enabled apps with Azure AI services	The endpoint and one of the
Designing and Implementing a Microsoft Azure AI Solution	Translate speech with the Azure AI Speech service	SpeechConfig
Designing and Implementing a Microsoft Azure AI Solution	Translate speech with the Azure AI Speech service	SpeechTranslationConfig
Designing and Implementing a Microsoft Azure AI Solution	Translate speech with the Azure AI Speech service	AudioConfig
Designing and Implementing a	Create an Azure Cognitive Search solution	Add a JSON file that defines container.

Microsoft Azure AI Solution		
Designing and Implementing a Microsoft Azure AI Solution	Create an Azure Cognitive Search solution	Enable anonymous access fo
Designing and Implementing a Microsoft Azure AI Solution	Create an Azure Cognitive Search solution	In an Azure AI Services resou references the container wh
Designing and Implementing a Microsoft Azure AI Solution	Create a custom skill for Azure Cognitive Search	Create a custom skill that us predict the sentiment for a c
Designing and Implementing a Microsoft Azure AI Solution	Create a custom skill for Azure Cognitive Search	Create a custom skill that ca predict the sentiment of eac
Designing and Implementing a Microsoft Azure AI Solution	Create a custom skill for Azure Cognitive Search	Add the built-in Sentiment sk
Designing and Implementing a Microsoft Azure AI Solution	Create a knowledge store with Azure Cognitive Search	Merge
Designing and Implementing a Microsoft Azure AI Solution	Create a knowledge store with Azure Cognitive Search	Shaper
Designing and Implementing a Microsoft Azure AI Solution	Create a knowledge store with Azure Cognitive Search	Split
Designing and Implementing a Microsoft Azure AI Solution	Enrich a search index using Language Studio	Conversational language und
Designing and Implementing a Microsoft Azure AI Solution	Enrich a search index using Language Studio	Analyze sentiment.
Designing and Implementing a Microsoft Azure AI Solution	Enrich a search index using Language Studio	Custom text classification.
Designing and Implementing a Microsoft Azure AI Solution	Implement advanced search features in Azure Cognitive Search	+

Designing and Implementing a Microsoft Azure AI Solution	Implement advanced search features in Azure Cognitive Search	^
Designing and Implementing a Microsoft Azure AI Solution	Implement advanced search features in Azure Cognitive Search	!
Designing and Implementing a Microsoft Azure AI Solution	Build an Azure Machine Learning custom skill for Azure Cognitive Search	Real-time endpoint
Designing and Implementing a Microsoft Azure AI Solution	Build an Azure Machine Learning custom skill for Azure Cognitive Search	Web service
Designing and Implementing a Microsoft Azure AI Solution	Build an Azure Machine Learning custom skill for Azure Cognitive Search	Batch endpoint
Designing and Implementing a Microsoft Azure AI Solution	Search data outside the Azure platform in Azure Cognitive Search using Azure Data Factory	You can only upload one doc
Designing and Implementing a Microsoft Azure AI Solution	Search data outside the Azure platform in Azure Cognitive Search using Azure Data Factory	The JSON can't contain comp
Designing and Implementing a Microsoft Azure AI Solution	Search data outside the Azure platform in Azure Cognitive Search using Azure Data Factory	You have to define the index
Designing and Implementing a Microsoft Azure AI Solution	Maintain an Azure Cognitive Search Solution	Create an Azure Cognitive S service tier and at least two
Designing and Implementing a Microsoft Azure AI Solution	Maintain an Azure Cognitive Search Solution	Create an Azure Cognitive S tier and at least three replica
Designing and Implementing a Microsoft Azure AI Solution	Maintain an Azure Cognitive Search Solution	Create an Azure Cognitive S tier and one replica.
Designing and Implementing a Microsoft Azure AI Solution	Use semantic search to get better search results in Azure Cognitive Search	Up to 50.
Designing and Implementing a	Use semantic search to get better search results in Azure Cognitive	As many results as the BM25

Microsoft Azure AI Solution	Search	
Designing and Implementing a Microsoft Azure AI Solution	Use semantic search to get better search results in Azure Cognitive Search	Up to 25.
Designing and Implementing a Microsoft Azure AI Solution	Improve search results using vector search in Azure Cognitive Search	To create a search to match
Designing and Implementing a Microsoft Azure AI Solution	Improve search results using vector search in Azure Cognitive Search	When you need to find matc search index.
Designing and Implementing a Microsoft Azure AI Solution	Improve search results using vector search in Azure Cognitive Search	To upload and index a docur
Designing and Implementing a Microsoft Azure AI Solution	Plan an Azure AI Document Intelligence solution	A Read model.
Designing and Implementing a Microsoft Azure AI Solution	Plan an Azure AI Document Intelligence solution	A Layout model.
Designing and Implementing a Microsoft Azure AI Solution	Plan an Azure AI Document Intelligence solution	A Composed model.
Designing and Implementing a Microsoft Azure AI Solution	Use prebuilt Azure AI Document Intelligence models	Read model.
Designing and Implementing a Microsoft Azure AI Solution	Use prebuilt Azure AI Document Intelligence models	General document model.
Designing and Implementing a Microsoft Azure AI Solution	Use prebuilt Azure AI Document Intelligence models	ID document model.
Designing and Implementing a Microsoft Azure AI Solution	Extract data from forms with Azure Document Intelligence	Train Model and Get Model L
Designing and Implementing a Microsoft Azure AI Solution	Extract data from forms with Azure Document Intelligence	Analyze Invoice and Get Ana

Designing and Implementing a Microsoft Azure AI Solution	Extract data from forms with Azure Document Intelligence	Create Azure Document Inte
Designing and Implementing a Microsoft Azure AI Solution	Create a composed Form Recognizer model	modelId
Designing and Implementing a Microsoft Azure AI Solution	Create a composed Form Recognizer model	status
Designing and Implementing a Microsoft Azure AI Solution	Create a composed Form Recognizer model	docType
Designing and Implementing a Microsoft Azure AI Solution	Build an Azure AI Document Intelligence custom skill for Azure Cognitive Search	formUrl
Designing and Implementing a Microsoft Azure AI Solution	Build an Azure AI Document Intelligence custom skill for Azure Cognitive Search	recordId
Designing and Implementing a Microsoft Azure AI Solution	Build an Azure AI Document Intelligence custom skill for Azure Cognitive Search	formSasToken
Designing and Implementing a Microsoft Azure AI Solution	Get started with Azure OpenAI Service	text-davinci-003
Designing and Implementing a Microsoft Azure AI Solution	Get started with Azure OpenAI Service	gpt-35-turbo
Designing and Implementing a Microsoft Azure AI Solution	Get started with Azure OpenAI Service	text-embedding-ada-002 (Ve
Designing and Implementing a Microsoft Azure AI Solution	Build natural language solutions with Azure OpenAI Service	Chat, Embedding, and Comp
Designing and Implementing a Microsoft Azure AI Solution	Build natural language solutions with Azure OpenAI Service	Key, Endpoint, and Deploym
Designing and Implementing a	Build natural language solutions with Azure OpenAI Service	Summary, Deployment nam

Microsoft Azure AI Solution		
Designing and Implementing a Microsoft Azure AI Solution	Apply prompt engineering with Azure OpenAI Service	By using complex instructior
Designing and Implementing a Microsoft Azure AI Solution	Apply prompt engineering with Azure OpenAI Service	By providing clear and descr
Designing and Implementing a Microsoft Azure AI Solution	Apply prompt engineering with Azure OpenAI Service	By using vague prompts
Designing and Implementing a Microsoft Azure AI Solution	Generate code with Azure OpenAI Service	Increase in efficiency and pr
Designing and Implementing a Microsoft Azure AI Solution	Generate code with Azure OpenAI Service	Increase in bugs and readab
Designing and Implementing a Microsoft Azure AI Solution	Generate code with Azure OpenAI Service	Increase in time spent codin
Designing and Implementing a Microsoft Azure AI Solution	Generate images with Azure OpenAI Service	DALL-E
Designing and Implementing a Microsoft Azure AI Solution	Generate images with Azure OpenAI Service	GPT-35-Turbo
Designing and Implementing a Microsoft Azure AI Solution	Generate images with Azure OpenAI Service	Text-Davinci
Designing and Implementing a Microsoft Azure AI Solution	Use your own data with Azure OpenAI Service	Create their own AI chat mo
Designing and Implementing a Microsoft Azure AI Solution	Use your own data with Azure OpenAI Service	Access Azure OpenAI withou
Designing and Implementing a Microsoft Azure AI Solution	Use your own data with Azure OpenAI Service	Use supported AI chat mode data

Designing and Implementing a Microsoft Azure AI Solution	Fundamentals of Responsible Generative AI	To make a legal case that in harms caused by the solutio
Designing and Implementing a Microsoft Azure AI Solution	Fundamentals of Responsible Generative AI	To document the purpose, e solution.
Designing and Implementing a Microsoft Azure AI Solution	Fundamentals of Responsible Generative AI	To evaluate the cost of cloud solution.

Your overall results

EXCELLENT

You are all set! Your results look strong and meet the necessary criteria for success.

CRITICAL 0-33 Critical: 0 to 33

MODERATE 33-67 Moderate: 33 to 67

EXCELLENT 67-100 Excellent: 67 to 100

Your result: 100/100 100 out of 100

Categories that influenced your results

Azure Machine Learning: Cost Optimization

EXCELLENT

Azure Machine Learning: Operational Excellence

EXCELLENT

Azure Machine Learning: Performance Efficiency

EXCELLENT

Azure Machine Learning: Reliability

EXCELLENT

Azure Machine Learning: Security

EXCELLENT

You can find out how to improve on individual categories by reviewing the recommendations below in the report.

Azure Machine Learning: Cost Optimization

What steps are you taking to optimize cloud costs in Azure Machine Learning?

We use cost management tools to plan and track costs.

We identify and use the right-sized compute for machine learning models.

We're considering the use of local compute or remote managed compute for automated machine learning to reduce costs.

We optimize our cloud investment with cost management.

We use an appropriately sized compute instance and compute cluster for training and inference.

None of the above.

How does your organization model and monitor cloud costs for Azure Machine Learning?

We plan and manage costs for Azure Machine Learning.

We utilize cost alerts for Azure Machine Learning.

We create and manage Azure budgets.

None of the above.

What actions do you typically take for cost optimization in Azure Machine Learning?

We configure our training clusters to autoscale.

We set quotas on our subscription and workspaces.

We set termination policies on our training job.

We use low-priority virtual machines (VMs).

We schedule compute instances to shut down and start up automatically.

We use an Azure Reserved VM Instance.

We train locally.

We parallelize training.

We set data retention and deletion policies.

We deploy resources to the same region.

How do you ensure that you have sufficient capacity?

We adhere to Azure Machine Learning subscription limits for compute and pipelines.

We review service limits in Azure Machine Learning.

We adhere to Azure Machine Learning subscription limits for storage.

None of the above.

Azure Machine Learning: Operational Excellence

How are you managing your machine learning lifecycle through automation using MLOps?

We use machine learning pipelines to orchestrate the workflow.

We use GitHub Actions with Azure Machine Learning.

We use MLFlow to track and manage ML modeling projects.

We use the Team Data Science Process for the data science project lifecycle.

None of the above.

What considerations are you making around the deployment of your infrastructure?

We segregate environments into development, test, and production workspaces.

We understand the Azure subscription limits that might impact this workload.

We use the Azure security baseline for Azure Machine Learning.

We set up separate workspaces for each publish environment.

None of the above.

How is development done on Azure Machine Learning workloads?

We use Python SDK or CLI to develop machine learning jobs.

We use low-code Azure Machine Learning designer to author experiments.

We use the Azure Machine Learning Notebook or Jupyter Notebook to author the experiment.

We use Azure Machine Learning automated GUI to author the experiment.

We use ONNX and deep learning libraries, such as Tensorflow, PyTorch, Keras, and others.

We use Responsible AI in our development practice.

None of the above.

How do you monitor your Azure Machine Learning deployments?

We monitor a deployed model by collecting and evaluating model data.

We enable logging in machine learning training runs.

We use alert rules and events in our application.

We analyze Azure Machine Learning platform metrics and logs from Azure Monitor.

None of the above.

How do you manage the configuration of Azure Machine Learning deployments?

We use code-based (SDK or CLI) definitions of our training jobs.

We use code-based definitions of our compute targets.

None of the above.

How do you test your MLOps infrastructure?

We use unit, regression, and integration testing with CI/CD for MLOps.

None of the above.

Azure Machine Learning: Performance Efficiency

How are you designing your Azure Machine Learning training workload to scale?

We use data partitioning strategy to run experiments in parallel, if possible.

We use autoscaling on clusters, where appropriate.

We use Azure Machine Learning pipeline step to process large amounts of data asynchronously and in parallel.

None of the above.

How are you designing your Azure Machine Learning service to meet performance requirements?

We use appropriate compute SKUs and sizes for different machine learning workloads.

We use the appropriate compute target types based on your workload requirements and environments.

We use datastore and dataset mounts for reusability throughout workload.

For unstructured files, we optimize performance by mounting data files to the compute target.

We use advanced automated ML options to increase performance/ROI on experiment run time.

How do you optimize data processing speeds for Azure Machine Learning workloads?

We set up Azure Machine Learning datastores/datasets to connect and access data from various storage services.

We use distributed training with Azure Machine Learning, where possible.

We use datasets/datastores to improve manageability, performance, and scale when working with data.

How do you monitor model performance and lifecycle activities?

We leverage Azure Machine Learning monitoring capabilities, such as model run logs and metrics.

We enable logging in Azure Machine Learning training runs.

We use Azure Monitor to monitor the performance of our model.

We leverage the Azure Machine Learning workspace job console to track workload progress.

How do you autoscale Azure Machine Learning compute resources to handle performance for training and inferencing?

We leverage Azure Machine Learning capabilities to autoscale the training compute nodes based on our benchmarking.

We leverage multinode scaling capabilities for model training.

We leverage production-grade model deployment and autoscaling inference using Azure Kubernetes Service cluster.

Azure Machine Learning: Reliability

What reliability considerations have you defined for your Azure Machine Learning workload?

We use a Managed Batch Endpoint for parallel batch processing.

We use a Managed Endpoint for scalable, self-managed service.

We use Azure Kubernetes Service (AKS) for high-scale production deployments with fast response time.

We manage and increase quotas for resources with Azure Machine Learning.

None of the above.

How do you ensure that your application architecture is resilient to failures?

We version and track Azure Machine Learning datasets.

We enable logging in machine learning training runs to support handling exceptions and errors.

We publish Azure Machine Learning components and environments.

None of the above.

What decisions have you made to ensure the application platform meets your reliability requirements?

We use scaling options for applications in Azure Kubernetes Service (AKS).

We use a managed endpoint for scalable deployments.

We manage a compute cluster in your Azure Machine Learning workspace.

We built a failover plan for business continuity and disaster recovery to respond to failures and disasters.

None of the above.

How do you monitor and measure both the health of a training run and the health of deployed service?

We collect machine learning log files in Application Insights.

We version and track Azure Machine Learning datasets.

We use native application monitoring.

None of the above.

What framework do you use to interpret ML models and help train unbiased models?

We check trained models for fairness.

We perform error analysis for trained models for reliability and safety.

We interpret trained models for transparency.

We perform causal analysis to understand how data impacts model decisions.

None of the above.

Azure Machine Learning: Security

What design considerations did you make in your workload in regard to security?

We use role-based access control (RBAC) to manage access to Azure Machine Learning workspaces.

We use Microsoft Entra ID for identity management and authentication of Azure Machine Learning users and processes for Azure Machine Learning resources and workflows.

We use MLOps practices for security guidance, model management, deployment, and monitoring with Azure Machine Learning.

We use appropriate recommendations for the Azure Machine Learning security baseline to improve security posture.

We review and implement appropriate guidelines from Azure Machine Learning best practices for enterprise security.

None of the above.

What considerations for compliance and governance have you made for your Azure Machine Learning workload?

We implemented a security and governance plan in accordance with guidance.

We audit and manage Azure Machine Learning using Azure Policy.

None of the above.

How do you manage encryption for workloads?

We use data encryption with Azure Machine Learning.

None of the above.

How do you manage identity for Azure Machine Learning workloads?

When running Azure Machine Learning workloads in Azure Kubernetes Service, we use Microsoft Entra Workload ID with Azure Machine Learning.

We use managed identities with Azure Machine Learning for access control.

None of the above.

How have you secured the network for your workload?

We use virtual networks (VNets) to secure an Azure Machine Learning workspace during setup.

We use virtual networks (VNets) to secure an Azure Machine Learning training environment.

We configured Azure Private Link for Azure Machine Learning to enable private endpoint for inferencing.

We secured an Azure Machine Learning inferencing environment with virtual networks (VNets).

We use Azure Machine Learning studio in an Azure virtual network (VNet).

We use TLS to secure web service through Azure Machine Learning.

None of the above.

How do you adhere to responsible ML principles in your design?

We use practices to protect users' data privacy in machine learning.

We work on encrypted data with homomorphic encryption.

We use model interpretability with Azure Machine Learning.

We assess fairness in machine learning models using open-source packages in Azure Machine Learning.

We perform causal inference on trained models.

Your overall results

EXCELLENT

You are all set! Your results look strong and meet the necessary criteria for success.

CRITICAL 0-2 Critical: 0 to 2

MODERATE 2-4 Moderate: 2 to 4

EXCELLENT 4-6 Excellent: 4 to 6

Your result: 6/6 6 out of 6

Categories that influenced your results

AVS | Readiness Resources

EXCELLENT

AVS | Marketplace Offer Development Resources

EXCELLENT

AVS | Specialization Resources

EXCELLENT

AVS | Cosell Acceleration Resources

EXCELLENT

You can find out how to improve on individual categories by reviewing the recommendations below in the report.

AVS | Readiness Resources

Opportunity and Use Cases

Migrating VMware vSphere workloads to Azure VMware Solution

Extending hybrid and multi-cloud agility

High availability and disaster recovery for VMware workloads

Desktop virtualization

Azure Migrate and Modernize and Azure Innovate

Training Resources

Introduction

Learning Path

Learning Resources

Overview Video

AVS Academy

VMware TechZone

VMware for Azure VMware Solution Master Specialist Exam

AVS LAB Automation

VMware AVS Hands-on Labs

AVS Workshop Lab Guide

Deployment Guidance

Landing Zone Accelerator

Landing Zone Accelerator GitHub Repository

Landing Zone Assessment Review

Landing Zone Assessment Network Design Guide

Deployment Checklist

Azure Well-Architected Assessment for AVS

Azure Well-Architected Documentation for AVS

Azure Proactive Resiliency Library for AVS

AVS Updates

AVS | Marketplace Offer Development Resources

Marketplace Training and Support Resources

Sell through the commercial marketplace

Plan a Consulting Service Offer, applicable for AVS Service

Partner Got-To-Market Toolbox

AVS | Specialization Resources

AVS Specialization details

Specialization Overview

Specialization Video

Specialization Audit Checklist

Specialization Assessment

AVS | Cosell Acceleration Resources

Go-To-Market Assets & Recommended Sellers Training

AVS Customer Story

IDC white paper: The Business Value of Azure VMware Solution

Digital Marketing Campaign (On Demand)

AVS Pricing Reference

AVS Go Big for Partners

AVS Partner Assets Collection

AVS Pros (LinkedIn Group)

Partners Incentives and Programs

AVS Bootcamp Sales Track

Azure VMware Solution (AVS) | Microsoft Partner - Mar 4, 2025 - 1:22:59 PM

Your overall results	Excellent	'6/6'
AVS | Readiness Resources	Excellent	'8/10'
AVS | Marketplace Offer Development Resources	Excellent	'3/3'
AVS | Specialization Resources	Excellent	'4/4'
AVS | Cosell Acceleration Resources	Excellent	'9/9'

Category	Link-Text	Link
AVS | Readiness Resources	Introduction	https://learn.m
AVS | Readiness Resources	Learning Resources	https://aka.ms
AVS | Readiness Resources	Learning Path	https://aka.ms
AVS | Readiness Resources	Overview Video	https://aka.ms
AVS | Readiness Resources	AVS Academy	https://aka.ms
AVS | Readiness Resources	VMware for Azure VMware Solution Master Specialist Exam	https://aka.ms
AVS | Readiness Resources	VMware TechZone	https://aka.ms

AVS | Readiness Resources	Landing Zone Accelerator	https://aka.ms
AVS | Readiness Resources	Landing Zone Accelerator GitHub Repository	https://aka.ms
AVS | Readiness Resources	Landing Zone Assessment Review	https://aka.ms
AVS | Readiness Resources	Landing Zone Assessment Network Design Guide	https://aka.ms
AVS | Readiness Resources	Deployment Checklist	https://aka.ms
AVS | Readiness Resources	Azure Well-Architected Assessment for AVS	https://aka.ms
AVS | Readiness Resources	Azure Well-Architected Documentation for AVS	https://aka.ms
AVS | Readiness Resources	Azure Proactive Resiliency Library for AVS	https://aka.ms
AVS | Readiness Resources	AVS Updates	https://aka.ms
AVS | Readiness Resources	Migrating VMware vSphere workloads to Azure VMware Solution	https://learn.m azure-vmware
AVS | Readiness Resources	Extending hybrid and multi-cloud agility	https://learn.m with-azure
AVS | Readiness Resources	High availability and disaster recovery for VMware workloads	https://learn.m vmware-site-r
AVS | Readiness Resources	Azure VMware Solution for Desktop virtualization	https://learn.m horizon
AVS | Readiness Resources	AVS LAB Automation	https://github
AVS | Readiness Resources	VMware AVS Hands-on Labs	https://aka.ms
AVS | Readiness Resources	AVS Workshop Lab Guide	https://aka.ms
AVS | Marketplace Offer Development Resources	Sell through the commercial marketplace	https://learn.m marketplace/
AVS | Marketplace Offer Development Resources	Plan a Consulting Service Offer, applicable for AVS Service	https://learn.m service-offer
AVS | Marketplace Offer Development Resources	Partner Got-To-Market Toolbox	http://aka.ms/
AVS | Specialization Resources	Specialization Overview	https://aka.ms
AVS | Specialization Resources	Specialization Video	https://aka.ms
AVS | Specialization Resources	Specialization Audit Checklist	https://aka.ms
AVS | Specialization Resources	Specialization Assessment	https://aka.ms
AVS | Cosell Acceleration Resources	AVS Customer Story	https://aka.ms
AVS | Cosell Acceleration Resources	IDC white paper: The Business Value of Azure VMware Solution	https://aka.ms
AVS | Cosell Acceleration Resources	Digital Marketing Campaign (On Demand)	https://aka.ms
AVS | Cosell Acceleration Resources	AVS Pricing Reference	https://aka.ms
AVS | Cosell Acceleration Resources	AVS Go Big for Partners	https://aka.ms
AVS | Cosell Acceleration Resources	AVS Partner Assets Collection	https://aka.ms
AVS | Cosell Acceleration Resources	AVS Pros (LinkedIn Group)	https://aka.ms
AVS | Cosell Acceleration Resources	Partners Incentives and Programs	https://aka.ms

AVS | Cosell Acceleration Resources ---	AVS Bootcamp Sales Track	https://aka.ms

Category	Question	Answers
AVS | Readiness Resources	Opportunity and Use Cases	Migrating VMw
AVS | Readiness Resources	Opportunity and Use Cases	Extending hyb
AVS | Readiness Resources	Opportunity and Use Cases	High availabil
AVS | Readiness Resources	Opportunity and Use Cases	Desktop virtua
AVS | Readiness Resources	Opportunity and Use Cases	Azure Migrate
AVS | Readiness Resources	Training Resources	Introduction
AVS | Readiness Resources	Training Resources	Learning Path
AVS | Readiness Resources	Training Resources	Learning Resc
AVS | Readiness Resources	Training Resources	Overview Vide
AVS | Readiness Resources	Training Resources	AVS Academy
AVS | Readiness Resources	Training Resources	VMware TechZ
AVS | Readiness Resources	Training Resources	VMware for Az
AVS | Readiness Resources	Training Resources	AVS LAB Auto
AVS | Readiness Resources	Training Resources	VMware AVS H
AVS | Readiness Resources	Training Resources	AVS Workshop
AVS | Readiness Resources	Deployment Guidance	Landing Zone
AVS | Readiness Resources	Deployment Guidance	Landing Zone
AVS | Readiness Resources	Deployment Guidance	Landing Zone
AVS | Readiness Resources	Deployment Guidance	Landing Zone
AVS | Readiness Resources	Deployment Guidance	Deployment G
AVS | Readiness Resources	Deployment Guidance	Azure Well-Ar
AVS | Readiness Resources	Deployment Guidance	Azure Well-Ar
AVS | Readiness Resources	Deployment Guidance	Azure Proactiv
AVS | Readiness Resources	Deployment Guidance	VMware Work
AVS | Readiness Resources	Deployment Guidance	AVS Updates
AVS | Marketplace Offer Development Resources	Marketplace Training and Support Resources	Sell through t
AVS | Marketplace Offer Development Resources	Marketplace Training and Support Resources	Plan a Consult
AVS | Marketplace Offer Development Resources	Marketplace Training and Support Resources	Partner Got-To
AVS | Specialization Resources	AVS Specialization details	Specialization
AVS | Specialization Resources	AVS Specialization details	Specialization
AVS | Specialization Resources	AVS Specialization details	Specialization
AVS | Specialization Resources	AVS Specialization details	Specialization
AVS | Cosell Acceleration	Go-To-Market Assets & Recommended	AVS Customer

Resources	Sellers Training	
AVS | Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	IDC white pap
AVS | Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	Digital Market
AVS | Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	AVS Pricing R
AVS | Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	AVS Go Big fo
AVS | Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	AVS Partner A
AVS | Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	AVS Pros (Link
AVS | Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	Partners Incer
AVS | Cosell Acceleration Resources	Go-To-Market Assets & Recommended Sellers Training	AVS Bootcamp

Goal

Start a new career

Build your plan

Take a few minutes to help us build your learning plan with AI—personalized to you!

- Objectives
- Timeline
- Review plan

Past 7 days

1. Mar 4

   You created a plan

plan

Develop AI Solutions with Azure AI Services (1)

- 4 items

2. Mar 4

You edited a plan

plan

Develop AI Solutions with Azure AI Services

- 4 items

3. Mar 4

You created a Collection

collection

Develop AI Solutions with Azure AI Services

- 8 items

4. Mar 4

You edited a plan

plan

Run data analytics solutions with Azure Databricks

- 9 items

5. Mar 4

   You started a plan

   plan

   Run data analytics solutions with Azure Databricks

   - 9 items

6. Mar 4

   You edited a plan

   plan

   Help secure your data in the age of AI (1)

   - 4 items

7. Mar 4

   You edited a plan

plan

Implementing data integration and model grounding with Azure AI Foundry and Microsoft Fabric

- 13 items

8. Mar 4

You started a plan

plan

Implementing data integration and model grounding with Azure AI Foundry and Microsoft Fabric

- 13 items

9. Mar 4

You edited a plan

plan

Build AI apps with Azure Services and best practices

- 20 items

10. Mar 4

You started a plan

plan

Build AI apps with Azure Services and best practices

- 20 items

11. Mar 4

You edited a plan

plan

Accelerate app development by using GitHub Copilot

- 23 items

12. Mar 4

You started a plan

plan

Accelerate app development by using GitHub Copilot

- 33 items

13. Mar 4

You edited a plan

plan

Help secure your data in the age of AI

- 4 items

14. Mar 4

You started a plan

plan

Help secure your data in the age of AI

- 4 items

15. Feb 28

You created a Collection

collection

46307064's Collection 9

- 0 item

Your activity feed shows up to 30 interactions over the past 30 days.

- Previo

g data issue rural system information management system education

46307064 0 Reputation points

Mar 4, 2025, 2:34 PM

isc tshingombe exam ims,, Access Control and Identity Management.pdf

Microsoft Authenticator

experimental career

by

tshingombe

editEdit gearManage timeHistory

Publication date

2025-02-03

Usage

Attribution-NonCommercial-ShareAlike 4.0 International

Topics

engineering

Collection

opensource

Language

English

Item Size

7.7M

this item is currently being modified/updated by the task: book_op

career experimental

Addeddate

2025-03-04 10:53:21

Engsecut

20000

Identifier

career-experimental-security-isc-microsoft-talk-brigth

Scanner

Internet Archive HTML5 Uploader 1.7.0

plus-circle Add Review

Reviews

There are no reviews yet. Be the first one to write a review.

0 Views

DOWNLOAD OPTIONS

download 5 files

COMMA-SEPARATED VALUES

download 12 files

PDF

download 1 file

TORRENT download

download 1 file

WEB VIDEO TEXT TRACKS download

download 2 files

WORD DOCUMENT

download 24 Files
download 24 Original

SHOW ALL

IN COLLECTIONS

Community Texts

Uploaded by Rdferz on March 4, 2025

13 reasons for seeking training and certification

Get started

We live in exciting times to be working in technology. While the words “digital transformation” can sometimes seem repetitive, we are indeed in a digital era, where so much of our technology is dramatically changing.

Pearson VUE surveyed more than 29,000 candidates who prepared for and earned IT certifications during the previous 12 months. Its resulting 2021 Value of IT Certification Report [1] features insights from North America, Latin America, Middle East and Africa (MEA), India, Greater China, Japan and the rest of Asia Pacific (APAC).

training and certification

© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet Ajet Ibraimoski

Senior Infrastructure/Systems Analyst Ajet Ibraimoski has used Cisco’s storehouse of knowledge to help

ace his CyberOps Associate certification exam and add to his know-how on the job. Cisco Packet Tracer,

online training, prerecorded videos, mentors, simulations, labs, Cisco Press books—Ajet has left no stone

unturned in his quest to understand the security field.

Certification benefits

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Knowledge: The systematic study of your field via a certification

program can help you understand it inside and out so that you are

more adept at attacking technical and business problems at many

levels to arrive at a solution.

> Get the full story on Ajet.

“Cisco is the biggest network vendor in the world, so if you want to learn networks, you turn to Cisco—with Cisco certification, you learn all different facets of security. It gives you an edge to design the right solution, right away—a solution that’s more secure, available, and scalable.”

—Ajet Ibraimoski, Senior Infrastructure/Systems Analyst© 2022 Cisco and/or its affiliates. All rights reserved.

Meet Randy Gates

Randy Gates left the security of a military career for the improved job prospects and flexibility of life as a

freelance network engineer. To realize this goal, Randy needed to ramp up his skill level fast. Picking up a couple

Cisco certifications got him up to speed—and now companies continue to call him back for freelance jobs.

> Learn more about Randy.

“The core knowledge from Cisco helps me expand my knowledge of routing and switching, IP subnetting, IP addressing schemes, firewalls, the Internet of Things, Power over Ethernet, and more. When I run into different problems, I know what to expect, and can give my customers a layout of what’s happening on-site.”

—Randy Gates, Network Engineer/Systems Engineer

Skill: A good certification program offers discipline and repeated

practice—with the structure to help you deeply ingrain a skill set,

and perform tasks more effectively and efficiently.

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

Certification benefits

© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet Jenny Guay

In search of her first job in the cybersecurity field, Jenny Guay didn’t have a lot of experience yet to show

her prospective employer in the interview, but there is one thing she did have: her CyberOps Associate

certification. It was sweet validation of her hard work to be able to answer questions with precision and to get the job.

> Learn more about Jenny.

“Cisco certifications shows an employer that you have time management and

project management skills, as well as the ability to complete a project that

you start. It looks impressive on a resume that you were able to study for

something challenging and then continue on to pass the certification exam.”

—Jenny Guay, Security Operations Center (SOC) Operator

Validation: Think of certification as a recognized badge of honor

that the rest of the world accepts and buys into.

Benefit 2 | Skill

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet Jose Bogarin

While studying electrical engineering at his university, he learned about Cisco certifications. Jose is now the

proud holder of 24 Cisco certifications, a member of DevNet 500 and the DevNet Class of 2020. Jose founded

his own IT services company that employs 60 professionals throughout Costa Rica and was named the Grand

Prize winner of Cisco's Platform Innovation Challenge. Jose continues to put customers first by developing

innovative IT technology.

> Explore Jose's path to success.

"You prepare, learn new concepts, take a tough but fair exam; then use the

certification to validate that you're ready to tackle all of the challenges a

particular technology can throw at you."

—Jose Bogarin, Chief Innovation Officer

Credibility: With validation goes credibility. Achievement via a strong certification program helps to establish with peers, managers, and customers that you know what you're doing.

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet DevonPatrick Adkins

When Devon was introduced to Cisco switches and wireless controllers, it was love at first touch. He realized

that all the super intelligent L2 and L3 engineers were Cisco certified and knew someday he would earn

his certifications. After earning his CCENT certification, he became the go-to guy for all things Cisco at his

company. And after receiving his CCNA certification, job offers started pouring in from around the world.

> Read on for more details about DevonPatrick.

“Cisco really changed my life and opened up a new world for me. The

certifications have helped me grow academically, mentally, and professionally.

I now see myself achieving quite a number of things in my life.”

—DevonPatrick Adkins, Network Engineer L2

Responsibility: We don’t mean responsibility as in “obligation”—

we mean the ability to take on added responsibility from your employer that will help you grow in your career.

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet João Muniz

Cisco certifications helped João build confidence to work in a real network with real issues. In addition,

it gave João's employer trust that he was well-trained for the job and had a strong work ethic and passion

for the profession.

> Get the full story on João Muniz.

"Cisco certifications give you the confidence that you know what needs to be

done in a real-world job. And knowing you are Cisco certified, employers are

confident that you will be successful on the job. They also know you have a

strong work ethic and are up for challenges, as they understand it's not easy

to get certified."

—João Muniz, Edge Data Center Project Engineer

Confidence: With the knowledge and increased skill gained from certification, and the rewards that accrue to that, comes increased confidence to do the job, which can build even further progress and self-assurance.

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet Kent Freeman

Originally an electrician, Kent Freeman’s true talent was in solving technology problems. So he transformed

his career, climbed the ladder, and ultimately launched his own IT services business. Kent has his CCNA,

his CCNP in the works, and his sights set on a CCIE—a direct reflection of his desire to push himself as well

as his business to continue evolving.

> Explore Kent’s path to success.

“You’re never too experienced to learn, to grow, to become better. And you’re

never too old to get certified. Technology is a rapidly moving industry. Getting

certified keeps you both relevant and competitive in the workforce.”

—Kent Freeman, Senior Collaboration Engineer

Value: All those good things like knowledge, skill, validation, ability to take on added responsibility, and confidence translate to your increased value as a successful performer in the eyes of employers and customers.

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

Certification benefits

© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet Keisha Richardson

At first, Keisha got certified because obtaining your certifications was required as part of the program she joined.

Once she got her first taste, however, she proceeded to study up and earn several additional certifications. She

likes the fact that certifications are a stamp of approval. While she may not have extensive experience that others

have, her certifications ensure she has the knowledge to discuss topics and tackle solutions.

> Read on for more details about Keisha.

“The time and effort that you put into earning a certification says more about

you as a professional than the certification itself.”

—Keisha Richardson, Technical Solutions Specialist

Versatility: A robust certification portfolio allows you to make

forays into different technology arenas, using those added skills

to enhance your current strengths and discover new interests.

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet Ronald Boestfleisch, Jr.

Eight years as a help desk rep had brought Ronald Boestfleisch's career to a standstill. But he was inspired to

become a network engineer and used a bachelor's degree and four Cisco certifications to propel himself into

his chosen field. It was a lot of hard work, but opportunity has knocked as a result. Ronald has now positioned

himself for senior network engineering roles.

Opportunity: Those who certify and gain new skills and knowledge

in the process stand a greater chance of breaking out of possible

job silos, taking advantage of cross-functional team roles, and

being in better position for an exciting career change.

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

> Get the full story on Ronald.

“Many employers offer incentives to employees who want to better themselves by getting certifications. That way, they gain employees who have more knowledge and skills, and really know what they’re doing. When I say I’m certified, employers’ ears perk up. It really helps with marketability and is a benchmark for a career.”

—Ronald Boestfleisch, Jr.,

Network Engineer 2, Managed Services Operations, Ops Control

Certification benefits

©

2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet John Warren

Diagnosed with multiple sclerosis (MS) in 2014, John struggles with an inconsistent memory, hand spasms,

delayed nerve responses and frequent debilitating exhaustion. Against all odds, Cisco certifications played

a key role in helping him become the guy a multinational behavioral healthcare provider relies on at night to

ensure their critical networks stay running.

> Read on for more details about John.

“I live with Multiple Sclerosis which is a very debilitating disease, however

I have overcome all adversity to achieve my dreams.”

—John Warren, Service Desk Analyst II

Hireability: Time and time again, we hear that certification

is one of those assets that employers automatically look for when it comes time to seek out new talent.

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Meet Aaron Dubin

Early on, Aaron knew that he wanted to expand his skills. But he didn’t have any mentors to guide him along,

telling him what to study and how to prepare. Through the certification process, he was able to get expert advice

on how to apply his particular interests, and where he should focus his time. He learned how to avoid going

down rabbit holes, and what type of preparation would ultimately help him the most in his career in technology.

> Learn more about Aaron.

“The value in certification is not in the piece of paper on your wall. It’s in the skills and the confidence you need to take on new challenges or take that next step in your career.”

—Aaron Dubin, Systems Engineer

Advancement: Being certified gets you noticed by your employer, who is more likely to remember your initiative when making decisions about your promotion.

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 12 | Salary

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

© 2022 Cisco and/or its affiliates. All rights reserved.

Meet Shane Woolf

Shane Woolf has a lot happening on the home front—he and his wife have built a blended family of two

biological children and six foster children. A good salary has helped him with that commitment, and he

credits his CCNA Routing and Switching and CCNA Collaboration certifications as helping to pay dividends

and get him to his current level as a Senior Unified Communications Systems Engineer.

> Explore further Shane's path to success.

"Cisco certifications helped me springboard my career to senior level—

with more responsibilities, a title, and a salary increase. And it helped

my employer, too, by giving them one more person to do the function

they're requesting."

—Shane Woolf, Senior Unified Communications Systems Engineer

Salary: As this list demonstrates, there is much more to certification than a fatter paycheck, but with advancement and career progression, more money can follow.

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 13 | Lifelong Learning

Benefit 1 | Knowledge

Certification benefits© 2022 Cisco and/or its affiliates. All rights reserved.

Cisco Public

Benefit 2 | Skill

Benefit 3 | Validation

Benefit 4 | Credibility

Benefit 5 | Responsibility

Benefit 6 | Confidence

Benefit 7 | Value

Benefit 8 | Versatility

Benefit 9 | Opportunity

Benefit 10 | Hireability

Benefit 11 | Advancement

Benefit 12 | Salary

Benefit 1 | Knowledge

Certification benefits

Each of the Cisco students we profiled has a different story, but they all have one thing in common: the will to

take that important first step toward advancing their career. That first step was earning a Cisco certification. With

this single act, they broadened their knowledge and skillset. They improved their understanding of networking.

They earned more respect from their peers. And they became more marketable.

Stake your claim to the future of technology—and the future you want to build for yourself—with Cisco certifications.

Your Cisco certification shows the world you have what it takes to work on today's most advanced networks.

Yet as technology evolves, so must the skills required to keep pace and maintain active certification. We call

this the journey of lifelong learning.

To learn more, visit Cisco Recertification Policy and Cisco Continuing Education.

The 2020 IT Skills and Salary Report from Global Knowledge

concludes that achieving a new certification, on average,

leads to a $13,000 salary increase.

Lifelong Learning: Since technology and innovation evolve so quickly,

it's important that you stay ahead of the latest advances and keep

your skills up to date. In the digital world, you can never automatically

assume that you're completely caught up and know everything—what's

new today can become outdated in seconds.© 2022 Cisco and/or its affiliates. All rights reserved.

Summary

We trust that at least some of the 13 reasons to get certified

have resonated with you.

Our digital era—with all of its excitement and change—dares

us to keep up with it. As many of the individuals highlighted in

this book have discovered, the pace of innovation in today's

technology arenas makes it difficult to earn just a single certification and then rest on one's laurels. Just as businesses must always stay innovative and one step ahead of the curve to thrive, so must you.

For many of us, the pace is motivating. Often the conquering of one certification challenge fires us up for the next one—part of a lifelong learning journey to be the best at what we do.

If you're ready to learn more, we have some ideas for you.

- Read Five Tips to Get DevNet Certified Now
- Get an overview of our certifications, register for these webinar series:
- CCNA Essentials
- DevNet Associate Essentials
- CyberOps Associate Essentials
- Learn about Guided Study Groups
- Join our communities on Cisco Learning Network and DevNet
- Amp your self-study plan with these programs:
- CCNA Prep
- DevNet Associate Prep
- CyberOps Associate Prep
- Cisco Expert Prep

Ask any of the certification holders we highlighted, and they'll tell you that training and certification transformed their careers. Their incomes. And their skill sets.

But beyond that, training and certification transformed their lives. By learning the latest methods and skills to connect, secure, and automate, they gained the self-confidence from setting goals, then accomplishing them. And that self confidence continues to empower them on their career paths.

We hope that training and certification empowers—and transforms—your career too.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks.

Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R) DEC21CS5955 01/22

References

1. 2021 Value of IT Certification Candidate Report, Pearson VUE

2.

2021 Value of IT Certification Candidate Report, Pearson VUE

3.

2021 Value of IT Certification Candidate Report, Pearson VUE

4. 2020 IT Skills and Salary Report, Global Knowledge

5. 2021 Value of IT Certification Employer Report, Pearson VUE

6. 2021 Value of IT Certification Employer Report, Pearson VUE

7. 2021 Value of IT Certification Candidate Report, Pearson VUE

8. 2021 Value of IT Certification Candidate Report, Pearson VUE

9. 2020 IT Skills and Salary Report, Global Knowledge

10. 2021 Value of IT Certification Candidate Report, Pearson VUE

11. U.S. Bureau of Labor Statistics, Occupational Outlook

Handbook (Updated: March 2021)

12. U.S. Bureau of Labor Statistics, Occupational Outlook

Handbook, Software Developers (Updated: September 2021)

13. 2021 Value of IT Certification Employer Report, Pearson VUE

14. 2021 Value of IT Certification Candidate Report, Pearson VUE

15. 2021 Value of IT Certification Candidate Report, Pearson VUE

16. 2021 Value of IT Certification Candidate Report, Pearson VUE

Summary

CCNA

Certification

Guide

Top Cisco resources to plan and

prepare for certification

Get startedTable of Contents

02 Overview

04 Certifications path

06 Vocabulary

10 Training options

11 Exam overview

12 Resources

13 Next steps

Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

01 CCNA certification overview

If you’re looking to embark on a rewarding

and lucrative information technology (IT)

career, obtaining your CCNA certification is a

great place to start.

Earning your CCNA gives you a solid

foundation for any field, role, or specialty you

want to pursue in IT. It covers the basics, from

IP addressing to security, automation, and AI.

A CCNA certification is the perfect start if you

know you want to build or support IT

infrastructure. You can specialize later.

A CCNA can help you prepare for

, includingC

B Network engineer`

B Network support technician`

B Network administrator`

B Business roles in IT organizations,

from sales and marketing to the
management track
a wide
variety of IT jobs

Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

02And while you build your skills, you’re also
building your income. Here’s proof. Skillsoft’s

Global Knowledge 2023 IT Skills and Salary

Report1 examined data from thousands of IT

professionals. Eighteen percent estimated

that the annual value of being certified is

$30,000 or more.

Training and certifications pay off— and

organizations know it. Over 35% of

respondents listed a boost in productivity as

the top benefit of a certified staff. Even more,

34% of respondents believe that certifications

close organizational skill gaps. That makes

certified candidates stand out in a pool of

applicants during the hiring process.

The network needs you. The field of IT is full

of rewarding, meaningful, challenging work.

Earning your CCNA certification can make your resume stand out and gets your foot in the door.

The CCNA arms you with a broad range of career skills. Get started today.

Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

7# Skillsoft, The 20 top-paying certifications going into 2024., https://www.skillsoft.com/

blog/top-paying-it-certifications, November 14, 2023.

Automation and programmability

IP services

Network access

Security fundamentals

IP connectivity

Network fundamentals

CCNA ExamOverview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

Look ahead – certification tracks

As you can see, CCNA is one of many steps you can take on your learning journey. With each step, you build your knowledge base—and your reputation—and become increasingly valuable to

any IT organization.

Download poster

Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

Cisco certifications are available

in multiple levels of expertise and various professional areas.

Earning the certification you need can lead you to the

career you want. It can also keep you competitive

in a field where 92 percent of IT professionals hold certifications. Cisco’s certification portfolio offers

more options than ever before, empowering you to customize your learning path to meet your career

needs, interests, and aspirations. Since every Cisco exam you pass earns you a certification, each of

these milestones you reach tells a new chapter in your story.

Here are the different levels of Cisco certifications you can earn:

Associate

Proof that you've mastered the

essentials to build your IT career

Expert

The most prestigious certification you

can obtain

Professional

A core technology track to sharpen

your specialized expertise

Specialist

Advanced networking knowledge in tech

such as security, data center, or video

Entry

Validate your skills and qualifications

for entry-level IT roles

Overview

Certifications path

Vocabulary

Training options
Exam overview
Resources
Next steps
©2024 Cisco and/or its affiliates. All rights reserved.

Vocabulary

Application Programming Interfaces (APIs)

and REST APIs

Attack surface

Authentication (Authentication, Authorization,

and Accounting [AAA], Radius)

APIs are published instructions to interface

with a product or service. APIs enable

developers to assemble a command or request

for a service or data, to submit it, and to

receive any output. They are published and

maintained by the vendor.

A collection of all the possible paths a hacker

or a malware application might follow to

compromise protected data.

Authentication is how you control access to

your network and prevent intrusions, data loss,

and unauthorized users.

Continuous Integration/Continuous

Development (CICD)

Data formats

A CICD system provides automated builds

and tests for creating software, making configuration changes, or completing other deployment tasks.When using a CICD pipeline, coders can continually merge their changes to a main branch of an existing application, run integration tests on changes, keep changes small, and minimize the

potential for problems due to multiple, gated

test result requirements.

(XML, JavaScript Object Notation [JSON],

YAML Ain t Markup Language [YAML])

Common data formats that are both machine

readable and human readable for providing

input to programs and applications using

interfaces (APIs).

Knowing these key terms will help you on your CCNA journey.

Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

DevOps

DNS

Infrastructure, containers, and

virtual machines

A combination of Development (Dev) and Operations (Ops), DevOps focuses on automation, regularly allowing failures that can be automatically fixed with mitigated risks, as well as connecting business outcomes to the availability goals for a given system. The DevOps movement makes developers

responsible for deployment and also has

teams use coding workflows and tools to

manage infrastructure.

The Domain Name Service (DNS) is like a

phone book that translates IP addresses into

human readable form. For example,

www.facebook.com is 157.240.22.35 (IPv4),

or 2001:558:feed::1.

Infrastructure is a generic term for the

underlying devices, physical or virtual, that

provide computing power or storage capacity

or networks, used to deliver software or

applications. Virtual machines can emulate a

computer system and are typically built as

images, providing the same functionality as

the physical computer. Containers package
up software and dependencies into one
descriptive file that contains everything
to run an application, regardless of the
underlying systems.
IP Addresses are like street addresses. Every
service or server on the internet has a unique
address where it can be accessed.

The process of determining the functionality,

origin, and potential impact of a given

malware.

IPv4 is limited to approximately 4 billion

unique addresses. NAT is a scheme that

allows a single address for a network (such

as a small business) to be shared by all the

users and devices on your network.

IP address (IPv4 and IPv6, classes, Open

Systems Interconnection [OSI] and TCP/IP

networking stack)

Malware analysis

Network Address Translation (NAT) Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

Network data models (YANG,

RESTCONF, NETCONF)

Packet

Python

Role-based access control

YANG is a data modeling language for

configuration and state data for network

devices. It stands for Yet Another Next

Generation. RESTCONF and NETCONF are

protocols defined by a standards body, so that

you can manage configuration of network

devices modeled with YANG.

A unit of data that can be sent from one

network endpoint to another. A packet has

headers, footers, and a data payload, or some

other information that it carries. The headers

encode details about how to route the packet.

A general-purpose, interpreted programming

language. Python emphasizes code readability

with whitespace requirements, so it is

approachable and powerful. Many network

automation applications and tutorials are

centered around Python.

Access to data given to a person based on

their job function or role.

Router

Routing protocols such as Border Gateway

Protocol (BGP), Enhanced Interior Gateway

Routing Protocol (EIGRP), and Open

Shortest Path First (OSPF)

Security Incident and Event

Management (SIEM)

A router connects different networks

together, providing a route between two

computers (or servers) in different networks.

Routers build the internet.

Routing protocols provide the overall map

and directions for a packet to find the

proper destination.

An approach to security management that

gathers data from multiple sources (such as

syslog, device events, and error logs),

processes the data (including correlation to

identify potential threats), and raises an alert

or ticket for further investigation if the threat is deemed to be real.Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

Security Orchestration and Automation Response (SOAR)

Software Development Kit (SDK)

Subnet

Switch

An approach that enables SOC teams to

manage tickets raised through SIEM (Security

Incident and Event Management) for threat

response. SOAR enables automated workflows

for responding to the threats.

A platform for writing programs and

applications targeting an API. It often includes

documentation, configurations, and tools (such

as compilers or linkers) to write and execute

the code to interface with the API.

Subnetting is a scheme for efficiently

apportioning or assigning your IP addresses to

systems in your organization.

A switch is a component that is used to build a

network and to connect hosts and servers

within a network. A switch cannot route

packets or data between networks.

Threat intelligence

Threat hunting

Time-based access control

VLAN

Evidence-based knowledge, including

context, mechanisms, indicators, implications,

and action-oriented advice about an existing

or emerging hazard to assets.

The process of proactively and iteratively

searching through networks to detect and

isolate advanced threats.

Temporary access to data given to a person

on a need basis for a period of time.

A Virtual Local Area Network is a simple

scheme to build in access control and restrictions within a network. It allows you to keep “Sales” separate from “Engineering,” for example, and to prevent inappropriate access to data.

Overview
Certifications path
Vocabulary
Training options
Exam overview
Resources
Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

Training options

There are several training options to help you prepare¼

Cisco Learning Locator

Cisco Press

CCNA 200-301 Official Cert Guide, Volume 2

CCNA Preparation Bundle

Implementing and Administering Cisco Solutions

| CCNA

Cisco Exam Review: CCNA

D Use the to find instructor-led

courses-both in-person and virtual.

D If books are your thing, check out the

, a

perfect addition to your self-study plan.

D The helps you prepare for

the 200-301 CCNA exam. For a limited time, save

16 percent off the bundle price.

D The

Learning Path will build the skills you need to

install, operate, configure, and verify basic networks,

while you prepare to take the 200-301 CCNA exam.2

D The helps you prepare to

take the CCNA exam. You will gain practical

experience for exam topics by testing your key

knowledge, skills, and abilities.

Regardless of how you prepare for the

exam, it s crucial to get your hands on

the gear to practice. This is called

labbing,” as in practicing in a lab
environment. Your ability to execute
critical tasks will be tested on the
exam, so you need to practice. Lab
early. Lab often. Then lab some more.
Explore Cisco Modeling Labs.
Keep this in mind: when the verb
for a topic area is describe,” you

won t need the same depth of knowledge for that topic as when
the verbs are configure,” troubleshoot,” and design.
This is where the real work happens. You ll need two things: the exam topics as well as a strategy for
learning, studying, and practicing.
The CCNA certification exam topics are the basis for the exam. They define the contents of both

the exam and the official training course, and it should be your roadmap for studying. If you can

successfully complete the tasks defined for each topic, you re ready for the exam.

Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

Exam overview

The CCNA exam topics are the basis for the certification exam. They define the contents of both the exam and the official training course, and should be your roadmap for studying. If you can successfully complete the

tasks defined for each topic, you’re ready for
the exam.
And you can take certification exams online,
so you can stay on track, even when you can’t
travel to a testing center.
200-301 CCNA Exam Topics
Visit
to perform a
system check.
www.cisco.com/go/
onlinetesting

CCNA certification exams are administered by our testing partner, Pearson VUE, as proctored exams.

When you take the exam, you'll be in a controlled environment to ensure fairness and to give you the

best, most consistent experience.

To view a walk-through demonstration

of the various exam question types and

how they function, check out the

page.

Cisco

Certification Exam Tutorial Videos

Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

©2024 Cisco and/or its affiliates. All rights reserved.

Resources

CCNA Essentials

Still thinking about getting certified?

Select CCNA Essentials for more details

about the program. It will help you make

the best decision.

CCNA Prep

Already decided to get certified, and want

to start studying? Select CCNA Prep to

find the ultimate resources to develop

your self-study plan.

Other CCNA resources include:

CCNA Certification

Download 200-301 CCNA Exam Topics

CCNA Certification Community

CCNA Certification Training Videos

Cisco Learning Blogs

Cisco Certification Blogs

Stay connected through:

LinkedIn

Facebook

X, formerly known as Twitter

Cisco Learning Network

Instagram

Use code CCNAGUIDE to save 35%

on CCNA 200-301 Official Cert

Guide Library at

www.ciscopress.com/ccnalibrary.

We have many resources to help your

progress. We encourage you to sign up for

the Cisco Learning Network to be able to

access learning resources, including videos,

learning plan and more. Just as an example

the CCNA Training Video learning plan has

over 52 hours of self study preparation tools

and content. Overview

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

Next steps

So many vibrant technological developments

culminating all at once can seem

overwhelming. You need to be able to

catch the wave of change and ride it smoothly.

This is precisely why we believe right now is

the best time to explore how training and

certification can keep you focused and

help make you a technical superstar in
your organization.
Earning a CCNA is the gateway to a rewarding
and lucrative IT career. With your CCNA, you'll
be more knowledgeable and confident about
all things IT.
Use this to let your manager know why training and certification is so
beneficial for you—and for them. Get the help

you need to transform your career, your

income, and your skill set.

email template

But beyond that, training and certification

can transform your life. By learning the latest

methods and skills in IT, you’ll gain the

self- confidence that comes from setting

goals, then accomplishing them. And that

self-confidence will continue to empower
your career.
These are exciting times to be working in technology. The IT landscape evolves daily, but
through it all, knowledge of routing and switching remain the core foundation on which you can
build your career.
Ask your manager to sponsor your
training. Once they understand the

benefits to your organization, they
will be eager to learn more and
support you with next steps.

Download e-book

©2024 Cisco and/or its affiliates. All rights reserved.

Certifications path

Vocabulary

Training options

Exam overview

Resources

Next steps

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and

other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/ trademarks. Third-party trademarks mentioned are the property of their

respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R) DEC22CS6087-1 08/24

So many opportunities not to miss

Because CCNA covers so many IT fundamentals, it’s a great way to stand out no matter

where your career takes you. Check out the following pages

to add more reasons for getting
your CCNA certification.
Land your dream role in tech. The
tech industry is growing rapidly,
adding hundreds of thousands of
new jobs a year. It’s a great time
launch or expand your career, and
no matter what inspires you, your

dream role is out there.

Learn how earning a Cisco

certification has positively changed

the life and career for others. Meet

the professionals who connect,

secure, and automate the world

while you find out the secrets to their

success. Get inspired! View

certification success stories.

Certifications matter. To employers
they indicate the promise of higher
productivity and proof of your
abilities. For new and current IT
professionals, they reflect far more.
Find out why.

With the rise in the adoption of AI and ML-related tools, many enterprises are defending their apps and infrastructure with automation. Some even leverage large language models (LLMs) to build their own AI-based solutions, managing associated apps and workloads. In this session, learn how F5 leverages AI and ML in our security products, including anomaly detection, malicious users, and more. By watching this session, your personal details will be shared with F5 and treated in accordance with the F5 Privacy Notice (https://www.f5.com/company/policies/privacy-notice). You also are signing up to be contacted about F5 products and services, but you can unsubscribe at any time.